Linear Algebra, Multivariable Calculus, and Modern Applications

Math 51 course text prepared by the
Stanford University Math Department

Last modified on July 12, 2024

Contents

Introduction . . . i
- Applications . . . vi
- Advice on studying, homework, and exams for math in college, and tutoring/online resources . . . vii
- Preparedness for Math 51 . . . x
- Advice to Instructors . . . xi

Part I. Geometry of vectors and projections . . . 1
- 1. Vectors, vector addition, and scalar multiplication . . . 3
- 2. Vector geometry in $\mathbf{R}^n$ and correlation coefficients . . . 21
- 3. Planes in $\mathbf{R}^3$ . . . 39
- 4. Span, subspaces, and dimension . . . 56
- 5. Basis and orthogonality . . . 71
- 6. Projections . . . 86
- 7. Applications of projections in $\mathbf{R}^n$: orthogonal bases of planes and linear regression . . . 100

Part II. Multivariable functions and optimization . . . 117
- 8. Multivariable functions, level sets, and contour plots . . . 119
- 9. Partial derivatives and contour plots . . . 133
- 10. Maxima, minima, and critical points . . . 152
- 11. Gradients, local approximations, and gradient descent . . . 173
- 12. Constrained optimization via Lagrange multipliers . . . 187

Part III. Geometry and algebra of matrices . . . 215
- 13. Linear functions, matrices, and the derivative matrix . . . 217
- 14. Linear transformations and matrix multiplication . . . 233
- 15. Matrix algebra . . . 254
- 16. Applications of matrix algebra: population dynamics, PageRank, and gambling . . . 263
- 17. Multivariable Chain Rule . . . 275
- 18. Matrix inverses and multivariable Newton's method for zeros . . . 295

Part IV. Further matrix algebra and linear systems . . . 317
- 19. Linear independence and the Gram–Schmidt process . . . 319
- 20. Matrix transpose, quadratic forms, and orthogonal matrices . . . 349
- 21. Linear systems, column space, and null space . . . 371
- 22. Matrix decompositions: QR-decomposition and LU-decomposition . . . 398

Part V. Eigenvalues and second partial derivatives . . . 431
- 23. Eigenvalues and eigenvectors . . . 433
- 24. Applications of eigenvalues: Spectral Theorem, quadratic forms, and matrix powers . . . 445
- 25. The Hessian and quadratic approximation . . . 467
- 26. Grand finale: application of the Hessian to local extrema, and bon voyage . . . 482

27. More eigenvalue applications: ODE systems, population dynamics, SVD (optional) 502

Appendices 519
A. Review of functions 520
B. Further details on linear algebra results (optional) 528
C. Equivalence of two perspectives on ellipses and hyperbolas (optional) 554
D. Google's PageRank algorithm (optional) 558
E. General determinants (optional) 561
F. The cross product (optional) 585
G. Neural networks and the multivariable Chain Rule (optional) 597
H. The QR algorithm (optional) 605
I. Newton's method for optimization (optional) 609
J. Hessians and chemistry (optional) 614

References 617

Introduction

“Linear algebra is the central subject of mathematics. You can’t learn too much linear algebra.”

B. Gross, former Dean of Harvard College

“Mathematical research can go farther and deeper than experiments ever will go. It can give *guarantees* that certain outcomes will *always* result or that certain outcomes will *never* result. When you first hear of such a guarantee, it can rock your world. [. . .] This got the attention of MRI researchers [. . .] Mathematics can also be a floodlight illuminating clearly the poorly understood path ahead.”

D. Donoho [Do, Sec. 3]

Why is linear algebra important? The 21st century is an age of data. In computer science, natural sciences, engineering, social sciences, and daily life, mountains of data are pervasive on a scale that would have been unimaginable at the end of the 20th century. Differential equations provided the mathematical framework for many of the advances of the 20th century, but linear algebra (the algebra and geometry of vectors and matrices in arbitrary dimensions) is the mathematical tool *par excellence* (alongside statistics) for the systematic analysis and management of the data-driven tasks of the 21st century. Even for modern applications of differential equations, linear algebra far beyond 3 dimensions is an important tool.

Stanford is unique among its peer universities in teaching “high-dimensional” linear algebra *alongside* multivariable differential calculus right from the start in its Math courses aimed at all students (not just future Math majors). You might wonder: why should I care about “high dimensions” (whatever that means), since we live in a 3-dimensional world? The reason is the following:

> *Many* contemporary real-world problems (in medicine, computer science, engineering, economics, physics, etc.) involve millions or even billions of variables that must be managed in a sensible way, and the only systematic way to handle this is via the language of high-dimensional linear algebra.

Here are some examples:

(i) A breakthrough in medical imaging as described in [Do], harnessed insights from linear algebra with 1,000,000-*dimensional spaces* (applied to 1000 × 1000 pixel arrays). In 2017 a further speed-up was brought to market, impacting millions of MRI scans annually.

(ii) Machine learning and AI systems involve real-world optimization problems in zillions of unknowns that must be solved in a reasonably short time. This relies on a lot of math, such as the multivariable Chain Rule for functions of *an arbitrary* number of variables, which can only be truly understood in an intuitive way via the framework of linear algebra we will teach you.

(iii) The crucial insight that made possible Google’s PageRank algorithm (and thereby so many subsequent webpage-ranking advances) was a synthesis of two fundamental concepts from linear algebra *taught in this course* – eigenvalues and Markov chains – applied to N-dimensional spaces with N on the order of **billions** (the total number of webpages on the Internet).

(iv) Google’s quantum computing breakthrough [Aru1] used random quantum circuits expressed as points in N-dimensional spaces with $N = 2^{53} \approx 10^{16}$ for the quantum mechanical analysis.

(v) The structural analysis of tall buildings and oil rigs (via the “finite element method”) leads to n equations in n unknowns for n in the thousands (or beyond). For the design of modern integrated circuits, Kirchhoff’s laws can amount to n equations in n unknowns with $n \approx 100,000$.

Topics we cover in linear algebra (e.g., Markov chains, eigenvalues) and multivariable optimization (e.g., gradient descent, least squares) pervade many applications. That is why CS229, CS230, and CS231N list this course as the math prerequisite: it is the *only* course at Stanford teaching **both** the linear algebra and optimization (“matrix calculus”) techniques used in many data analysis and machine learning courses. It also provides **all** of the linear algebra background for EE263, useful in engineering and economics.

The fundamental insight of single-variable calculus is that any (reasonable) function $f(x)$ is approximated near $x = c$ by a function of the form $ax + b$ (where $a = f'(c)$). The fundamental insight of multivariable calculus is that any (reasonable) function $f(x_1, \ldots, x_n)$ of n variables is approximated by a multivariable generalization $A\mathbf{x} + \mathbf{b}$, where now $\mathbf{x}$ and $\mathbf{b}$ are *vectors* and A is a *matrix*. **To make sense of this, we need linear algebra.**

> In this course, we develop linear algebra because this language expresses the ideas of multivariable calculus in a manner that provides visual insight even for n-variable problems with $n > 3$ and *looks very similar to single-variable calculus* in its language.

You may be surprised at how much emphasis we place on linear algebra: approximately 2/3 is linear algebra, and 1/3 is calculus (in the derivative aspects, which pervade all quantitative fields). Introducing linear algebra at the outset allows us to develop multivariable calculus very efficiently and to understand where concepts come from and (at least informally) why things work as they do. Moreover, *linear algebra is fundamentally important in its own right!* Multivariable calculus is crucial in the natural sciences (e.g., physics, engineering), economics, and machine learning, but linear algebra arises across even more fields, and provides an essential framework for statistics, biology, and data science. By learning multivariable calculus with insights from theorems (not just language) of linear algebra, **you** can do even more.

Why haven't computers made human understanding of math obsolete? Knowledge of the linear algebra in this book will give *you* "geometric intuition" for problems involving a large number of variables, as arise in practical problems in many fields. The ability of AlphaZero to rediscover (and go beyond) in a matter of hours all human knowledge in chess and Go acquired over thousands of years is a dramatic illustration of the power of neural networks [**SH**]. But human creativity and intelligence remain as valuable as ever because life is not a 2-player complete-information board game: AI systems are easily fooled [**He**], susceptible to bizarre flaws, and lack both scientific creativity [**DM**] and true intelligence.

> The widespread use of computers makes it *more important*, not less, for users of math to understand concepts. It is not necessary to know proofs, but novel users of quantitative tools are those with an intuitive understanding of ideas and how they fit together with applications and examples.

Many (especially online) courses enable people with little or no knowledge of math to apply powerful existing software to datasets. Those courses want to make everything look as simple as possible, so they teach how to use packages such as Tensorflow and sklearn that make it very easy to use a black box hiding all of the difficulties. That is "applied math" truly in the past tense: applying math that other people have already done so that you can solve the same problems they did but with different numbers. We will teach you ideas alongside techniques and motivation so you can be confident when "applying math": using it as a creative tool to solve new problems. Algorithms in the real world don't always work as desired, and then figuring out what to do requires knowledge of the underlying math and how concepts relate to each other.

A solid command of linear algebra and multivariable optimization will enhance your ability to turn informal ideas into working products and give you a substantial creative edge over those who only know how to plug into black-box formulas without real conceptual understanding. We will teach you the tools to handle many mathematical problems and concepts, illustrated with examples in modern contexts.

We put less emphasis on extensive bare-hands computation and more emphasis on matrix algebra as well as awareness of contemporary applications and the role of mathematical software when appropriate to solve problems. But just as practice with bare-hands arithmetic using small numbers in elementary school gave you some "number sense", a modest amount of bare-hands computation helps to aid your understanding when encountering new ideas. Hence, we do ask you to compute some examples by hand without getting too messy, and *knowledge of programming is not at all necessary for this course*. Please pass along any corrections, typos, etc. to math51book@lists.stanford.edu.

How you should read this book. Anything in a blue box (or purple or lavender in some PDF viewers), such as the following, is a core mathematical result or concept. Do not skip it, no matter what.

> Here is a very important result. Please read and understand me.

Such boxed text will make more sense if you read the surrounding material, so don't ignore everything else written there. Study worked examples to see what new concepts and results mean in specific situations. As a **study guide**, at the end of each chapter is a 1-page summary of notation, concepts, results, and skills.

In mathematics, a result that is important is often called a "theorem". We label such results in this way:

Theorem. The solutions of $x^2 + bx + c = 0$ are $x = \dfrac{-b \pm \sqrt{b^2 - 4c}}{2}$.

The word "theorem" should not be confused with the word "theory"; a theorem is a mathematical fact which has been proved to be true, and (in contrast with how theories develop in some other fields) *a theorem in mathematics does not become false later on as new ideas are discovered.*

A less important (even if very useful) result is sometimes called a "proposition":

Proposition. If a and b are numbers then $a + b = b + a$.

A consequence of a theorem or proposition is sometimes referred to as a "corollary", such as this:

Corollary. If $b^2 - 4c < 0$ then $x^2 + bx + c = 0$ has no solutions in the set $\mathbf{R}$ of real numbers.

In this course we will *never* ask you to prove a theorem (or proposition or corollary). Sometimes we provide an informal explanation for why a result is true, when we feel that some awareness of the explanation can aid in one's understanding (*but you won't be tested on reproducing such arguments*). We have included optional sections, typically near the end of chapters, which provide more details about justifications for those who are interested in such things; these can be safely ignored by those who are not interested (i.e., an "optional" label truly means what it says, and there is no extra credit for the study of the optional material). The end of an Example is denoted with ■. When we give a proof of a result (typically in an optional section), we often denote the end of the proof with the symbol □; you can ignore proofs!

Especially in the calculus portions of this book, there is a lot of "legal language" associated with the mathematics. For example, in order to differentiate a function, it must satisfy some technical conditions. We have tried to suppress this type of discussion whenever possible. (If you want to learn the subject more deeply, you will have to come back and learn some of those things, such as by reading optional material in this book and/or taking further Math courses.) **Material that is skippable and only of interest for technical completeness appears in a gray box so you don't feel overly tempted to read such things** (but it may appeal to those seeking a deeper understanding). Here is a sample (which you can ignore).

> Einstein discovered that the underlying geometry of the universe on a global scale is not that of Euclidean geometry as taught to students everywhere for more than 2000 years, but rather is a different kind of geometry introduced by Riemann in the 19th century. However, the theorems of Euclidean geometry remain *as valid as always* within their own framework and are used *every day* in modern data analysis techniques and so much more. The applicability of a single mathematical theorem to a wide array of different-looking contexts is possible precisely because mathematical truth depends only on logical reasoning from precise definitions and not on a specific application, though certainly applications provide useful motivation for and intuition behind the mathematics.

Terminology and notation. We often use the phrase "the following are equivalent", like this:

Theorem. For a quadratic polynomial $f(x) = x^2 + bx + c$, the following are equivalent:
(i) $b^2 - 4c > 0$.
(ii) the equation $f(x) = 0$ has two different solutions in $\mathbf{R}$;

(iii) the parabolic graph of $y = f(x)$ passes below the x-axis;

What this means is: *if any of (i), (ii), or (iii) holds for some specific* b *and* c *then the other two hold as well* (and so correspondingly, if even one of them is false for specific b and c then all are false for that b and c).

Let's see why those statements (i), (ii), and (iii) are equivalent to each other. Suppose (i) is true, so $b^2 - 4c > 0$. Then the equation $f(x) = 0$ has two different solutions in $\mathbf{R}$ because the quadratic formula $\dfrac{-b \pm \sqrt{b^2 - 4c}}{2}$ consists of two different real numbers, and likewise the parabola $y = f(x)$ then cuts through the x-axis at those two real roots. So if (i) is true then so are (ii) and (iii).

On the other hand, when (ii) is true (i.e., $f(x) = 0$ has two different solutions in $\mathbf{R}$) then by inspecting the output of the quadratic formula the values $\pm\sqrt{b^2 - 4c}$ are different real numbers and so the real number $b^2 - 4c$ must be positive. This says that if (ii) is true then so is (i), and hence so is (iii) (since we have seen already that (iii) holds whenever (i) does).

Finally, if (iii) holds then the parabola $y = f(x)$ that is largely above the x-axis (since the leading coefficient of the quadratic polynomial $f(x)$ is positive) must cross the x-axis at two different points (one on the way down, the other on the way up) and hence (ii) holds. We have already seen that when (ii) holds then so does (i).

So if one of (i), (ii), or (iii) is true (for a given b and c) then all are true (for the same b and c).

We need some notation for efficiency of language. Here is a summary table (in which the final symbol refers to points of space much as $\mathbf{R}^2$ refers to points in a plane with an origin and coordinate axes):

When you see the symbol ...	read it as
$\{\ \}$	"the set of " (or "the collection of")
$\subset$	"is a subset of"
$\in$	"belongs to" or "belonging to" or "is an element of"
: or |	"for which" or "such that"
$\sum_{i=1}^{10} c_i$	"$c_1 + c_2 + \cdots + c_{10}$"
$f : A \to B$	"f is a function taking input from A and producing output in B"
$\mathbf{R}^3$	"the collection of triples (x, y, z) of real numbers"

TABLE 0.0.1. Some useful notation

Here is an example: $(1, -3, 2) \in \{(x, y, z) \in \mathbf{R}^3 \,|\, x + y + z = 0\}$. Using the translation guide above, this turns into the following sentence: $(1, -3, 2)$ belongs to the set of all (x, y, z) in $\mathbf{R}^3$ for which $x + y + z = 0$. And this is true just because $1 + (-3) + 2 = 0$. Another example is:

$$\text{The function } f : \mathbf{R} \to \mathbf{R} \text{ defined by } f(x) = \sum_{j=0}^{n} x^j \text{ is equal to } \frac{x^{n+1} - 1}{x - 1} \text{ if } x \neq 1.$$

This asserts the formula for computing a finite geometric series. And for one more example:

$$\{x \in \mathbf{R} : x = y^2 + y \text{ for some } y \in \mathbf{R}\} \subset [-1, \infty). \qquad (*)$$

This says that a real number x that can be written in the form $y^2 + y$ for a real number y must belong to the interval $[-1, \infty)$ of real numbers greater than or equal to -1. In other words, this says $y^2 + y \geq -1$ for all y, or equivalently $y^2 + y + 1 \geq 0$ for all y. (In fact, numbers of the form $y^2 + y$ are exactly those at least $-1/4$, so the containment of sets in (*) above leaves room for improvement at the left endpoint.)

Applications

In the main text, discussions related to significant real-world applications (which can be looked at or ignored, according to your interests) are generally indicated by a green box, like this:

> The singular value decomposition of matrices, a consequence of the Spectral Theorem and orthogonal complements, is one of the most important concepts in machine learning. It is used in robotics, gene expression analysis, image compression, ridesharing, web search, quantum information, ...

Below is a table of *some* of the applications discussed in the book, to give you a sense of the wide range of utility of the mathematical techniques that we develop. There are many additional application contexts mentioned in the book for motivational purposes. A good understanding of these applications requires a solid grasp of numerous general concepts covered in the course.

Math topic	Some application(s)	Location in course text
$\mathbf{R}^n$ for big n	genomic analysis, MRI, massive data	page 1 of Intro., Ex. 1.2.3
Distance in $\mathbf{R}^n$	clustering algorithms	Ex. 1.6.13
Angles in $\mathbf{R}^n$	document/protein similarity, correlation	Ex. 2.2.4, Sec. 2.4
Pythagorean Theorem in $\mathbf{R}^n$	error estimation via correlation	Sec. 2.3, (7.3.5), Sec. 7.5
Projection to a line	balance of forces, quantum mechanics	Ex. 6.1.3, Ex. 6.1.6
Projection to a subspace	modern portfolio theory, support vectors	Ex. 4.1.9, Rem. 19.3.9, Ex. 19.4.4
Linear least squares	computing best-fit line to big data	Ch. 7, Ex. 10.3
Functions $\mathbf{R}^n \to \mathbf{R}^m$	Marshallian demand, mechanics, SVM	Ex. 2.1.10, Ex. 13.5.7, Sec. 24.5
Non-linear least squares	energy minimization, curve fitting	Exs. 10.1.3, 10.3.2
Linear programming	extrema in economics, strange diets	Ex. 12.4.2, Exs. 12.4.4, 13.3.11
Gradient descent	machine learning, physics, bio., chemistry	Sec. 11.3, Sec. G.3
Lagrange multipliers	constrained extrema (econ., entropy, SVM)	Ch. 12, Ex. 19.4.4
Interior point methods	extrema for very many unknowns	Ex. 12.4.4
Derivative matrix	Kalman filter, materials science, robotics	Exs. 13.5.10, 13.5.7, 18.5.7, 24.6.7
Affine functions	computer visualization, linear classifiers	Exs. 14.1.4–14.1.6, Ex. 19.4.4
$AB \neq BA$ for matrices	quantum mechanics, computer science	Rem. 14.3.6, Rem. D.2.1
Markov chains	gambling, pop. dynamics, PageRank	Ch. 16, Rem. J.2.1
Chain Rule	backpropagation in neural networks	Ch. 17, App. G
Matrix inverse	Newton's method (GPS, graphics), robotics	Sec. 18.5, Ex. 18.5.7
Matrix transpose	quantum comp., efficient curve-fitting	Rem. 20.1.11, Secs. 20.6, 20.8
Orthogonal matrices	modeling rigid motions, robotics	Sec. 20.4
Gram–Schmidt	signal processing, finance	Rem. 19.2.1, Rem. 19.3.9
Linear systems	electronics, input-output model in econ.	Sec. 21.1
Column/Null space, rank	overfitting, Netflix algorithm	Secs. 21.4, 21.6
QR-, LU-decompositions	high-dim. least squares, Cholesky decomp.	Ex. 22.5.3–Ex. 22.5.5, Ex. 26.1.10
Eigenvalues	long-term dynamics, PageRank	Secs. 24.4, 27.1, 27.2; Rem. D.2.1
QR algorithm	computing eigenvalues for many fields	App. H
Quadratic forms	rigid-body mechanics, thermodynamics	Rem. 20.3.13, Exs. 20.3.14, 24.2.7
Spectral Theorem	rot. mechanics, material stress, SVM, SVD	Secs. 24.2, 24.5, 24.6, 27.3
Higher partial derivatives	Partial diff. eqns in all scientific fields	Ex. 25.1.2
Hessian	optimization, robotics, molecular structure	Rem. 26.1.6, App. I, App. J
Singular value decomp.	rank reduction, PCA, genetics/eigengenes	Ex. 27.3.8, Rem. 27.3.9, Ex. 27.3.11
Polynomial interpolation	cryptography	Rem. E.5.6

Advice on studying, homework, and exams for math in college, and tutoring/online resources

"Mathematics *is* a language."

J.W. Gibbs, inventor of vector calculus

"First you learn how to work the problem, and then you can go use the computer."

Katherine Johnson, NASA mathematician

"Suppose that you are given a problem to solve ... a very powerful approach to this is to attempt to eliminate everything from the problem except the essentials; that is, cut it down to size. ... if you can bring this problem down into the main issues, you can see more clearly ... and perhaps find a solution."

Claude Shannon [**Shan2**]

Study and work habits

(a) Work with others, early and often! Even if you went through high school doing math largely on your own, there is no reason to stick with that work habit. Explaining things to classmates is a good way to identify gaps in your own understanding.

(b) Math is cumulative, so don't let confusion pile up: ask questions in office hours, tutoring, class, to classmates, etc. Asking for help is strongly encouraged. Do *not* blindly memorize things! If the motivation for a definition or the meaning of a result or method is unclear, ask about it.

(c) Math can only be learned actively. There is almost no value to looking over notes (or a textbook, or watching a video) and thinking "oh, I can do that; it makes sense". Watching someone else do a problem, or reading a solution, is wildly different than facing the problem on a blank sheet of paper. One can't learn to swim by just reading a book, or learn to build a tiled hut from sticks and stones just by watching this video, and likewise math cannot be learned by only watching others do it. In math, all reading must be done with paper and pen/pencil in hand. Instead of just reading worked examples, actively write out solutions on your own.

(d) Math is about ideas as much as it is about computations and special cases; this is what makes it useful in many different fields. Homework is essential to learning, since it is your opportunity to use concepts in both familiar and new settings; the latter (not emphasized enough in high school) yields robust understanding.

(e) Build up understanding of ideas first in a special case before the general case. After class, read about the material in the book, reread things multiple times, and look through the highlight summary at the end of the chapter. After reading the statement of a result, work out what it is saying in some special cases (e.g., low dimensions, three vectors instead of k vectors, etc.).

(f) When thinking about a worked example, make sure *all* steps make sense and are consistent with everything else (there are *no* inconsistencies in mathematics). If you encounter *any* inconsistency or something makes no sense even as a statement (setting aside what its derivation might be), sort out the confusion right away or make a note to come back to it soon.

(g) Use special cases and low-dimensional pictures to remember things. Remembering an idea or a visualization is often easier than memorizing a formula, and is a more reliable kind of knowledge.

(h) Read solution sets (for homework and worksheets) in detail, even for problems that you solved correctly; if a solution set uses a different method than you did or expresses the same method differently, try to understand it. If you get a correct answer but don't understand what you're doing, talk to someone to understand it.

(i) More than in perhaps any other field, definitions are absolutely crucial in mathematics (not emphasized much in high school). Read definitions very carefully. Make sure the syntax makes sense, terms make sense as written, and that they "work" on some examples you make in special cases (plug in numbers, etc.).

(j) Reread definitions, try them in special cases. If there are too many variables, plug in small numbers to see what something is saying. However, sometimes structure is rendered invisible after computing too explicitly; e.g., the factorization $9373 = 103 \times 91$ is not obvious, but when written as $97^2 - 6^2 = (97+6)(97-6)$ we recognize the rule for factoring a difference of squares.

(k) If something seems meaningless, go back to the definitions (or check hypotheses of results being used) and try to express it in a special case.

Homework and exams in college-level math classes

Homework is an essential part of your learning. Copying someone else's homework or not trying to complete the homework will hurt your understanding of the material and consequently your grade (and your ability to apply the material elsewhere). Treat homework as your opportunity to improve your skills and practice in preparation for exams. Homework is also how you become better at mathematics.

Homework will have problems that you have not seen in class. In contrast with high school, the goal of the course is not to teach you how to repeat previously seen patterns; it is to give you an understanding of concepts so you can apply them with confidence to a variety of scenarios, as well as new circumstances later in other fields. In class you'll be shown some ideas and results, and how to think about them, and some examples. But homework is by far the best way for you to identify gaps in your understanding and to internalize how ideas fit together. A good analogy is that we give you a hammer and nails/screws and show you how to build a table, but then ask you to build a chair at home.

Homework problems can sometimes be harder than the problems done in class. College math classes explain main ideas and results, discuss motivation behind them, and show how to use them. Homework is intended to challenge your understanding of the material. Only by being pushed beyond your level of comfort can you truly understand the concepts and broaden your mathematical skills.

There will be homework problems that you will not know how to do immediately. Getting stuck on problems or being confused about the material is a normal part of a college math class. If you never get stuck and always know what to do instantly, you are probably wasting your time in this class. If you get stuck, and remain so after mulling things over in your mind, ask for help.

Exams will include questions that you have not done in class or in homework. The goal of exams is to probe if you understand the concepts in addition to applying procedures, not to confirm that you can memorize previously-seen examples. Some exam questions may closely resemble those seen in homework or class, but others may check your understanding of the material by using it in a new context.

For good exam preparation, work through practice exams *without* the solutions in hand. Check your understanding via worked examples in the book, practice problems, and homework solutions. If you go through practice exams passively by reading solutions rather than first working on things by yourself, you will gain no benefit and will badly misjudge the level of understanding measured in exams.

The value of your Stanford degree reflects an expectation that you can handle new circumstances. Parroting procedures to solve known problems is often sufficient for success in high school math classes, but not here and not in future careers. Struggle is good in math (as in sports and music); it is how you improve your persistence and problem-solving abilities.

Tutoring

Here are some free tutoring resources for Stanford students that previous Math 51 students have found to be helpful.

- CTL tutoring (with many tutors available in math)
- Academic Coaching is a free resource on managing many skills related to academic success.
- AARC drop-in and group tutoring for student-athletes

Computational software (no coding needed)

- WolframAlpha for checking matrix calculations, contour plots
- Reshish is a matrix calculator, useful with: matrix inverse, powers, transpose, multiplication
- Matrixcalc for matrix calculations
- Desmos for 2d graphs of level curves $f(x, y) = c$ (or equations $f(x, y) = g(x, y)$).
- CalcPlot3D for dynamic 3d surface graphs $z = f(x, y)$ (enter $f(x, y)$)
- Geogebra for dynamic 3d surface graphs $z = f(x, y)$ (enter $f(x, y)$)

Internet video and website resources

Please *be aware* that many online video tutorials and website written material deviate from this book: they may develop material in a different order, rely on concepts or techniques that we do not use, and/or formulate important definitions or results in a more abstract or less useful manner than we do.

Two guiding principles in the writing of this book were to give (i) concrete definitions and numerical experience before more conceptual perspectives (without diminishing the importance of the latter), (ii) concepts, techniques, and results that are applicable uniformly to $\mathbf{R}^n$ for all n (important for many contemporary applications of linear algebra and multivariable optimization across many fields).

By the end of the book *you will know the standard main concepts and results* (and beyond!), but the route to get there is not the traditional one. The approach used is designed to provide a more confident and applicable command of the material than the traditional order in which it is taught. So *please be cautious when using other resources* (e.g., MIT OpenCourse videos for linear algebra).

- 3blue1brown has a popular video series Essence of Linear Algebra giving excellent dynamic visual illustrations (the mastermind behind the website is a former Stanford Math major)
- Khan Academy has linear algebra videos with some foundational deviations from how topics are developed in this book, but they are generally useful anyway.
- PatrickJMT is a huge collection of math videos, including a large section on linear algebra, of which the most relevant ones are on matrix multiplication (here, here, and here) and applications of Markov matrices (here and here).

  Beware that for some other topics, the methods used in the PatrickJMT videos are conceptually very different from the methods in this book. For example, the videos on Lagrange multipliers are essentially pure algebra, without any discussion of the role of gradient vectors.
- Paul's Onine Notes for geometry in $\mathbf{R}^3$ (with lines, planes, and vectors), as well as partial derivatives and applications thereof. Each section includes practice problems with solutions.

Preparedness for Math 51

"We choose [to do things] not because they are easy, but because they are hard, because that goal will serve to organize and measure the best of our energies and skills."

John F. Kennedy

Please take seriously the guidance from the placement diagnostic, and use it as an opportunity to meaningfully assess your background (especially to identify any rusty areas that you can reinforce with the recommended video reviews and practice problems). If the diagnostic indicates that you are not yet ready for Math 51, please do not disregard its recommendations. Here are some reasons that Math 21 is an enforced prerequisite for taking Math 51:

(1) Math 21 material arises in essentially all quantitative areas: differential equations (Math 53), probability & statistics (so machine learning and data-intensive experimental scientific work), finance & economic models, efficiency estimates in CS, etc. A former Math 21 student said:

> "there is no hope of you being a minimally decent engineer/physicist if you don't know how to do any of these things."

This applies equally well to future economists, chemists, biologists, and computer scientists.

(2) A Stanford student who did well in math in high school by rote learning once wrote:

> "my foundational math skills were poor and I struggled [with] new material [in other fields] more than I should have. [. . .] I [spent] a summer break working through calculus books to make up for it. Don't make the same mistake I did: put in the time to learn the material properly. It will pay for itself many times over."

The purpose of required math courses in a major is not merely to fulfill a requirement. It is to give you the skills to apply what you have learned in a confident and reliable manner. It is not about blasting ahead as quickly as possible. Short-term thinking does not serve long-term goals.

(3) Math 21 homework gives students practice writing more complex solutions than they ever see in Math 20's syllabus (AB-level AP calculus). Those higher-level skills are similar to what arises in Math 51, even if the content is not directly encountered in Math 51. So the content of Math 21 improves one's overall thinking about math, which is relevant even in circumstances where the Math 21 content is not.

(4) For those who take Math 20: the demands of a 5-unit course (such as Math 51) are a significant ramp-up from a 3-unit course (such as Math 20), especially if the latter was at least a partial review for the student, since then the student had a cushion (in Math 20) that will not exist for them in the succeeding course. So it is a more gentle adjustment to go through Math 21 (a 4-unit course) before Math 51.

You will go through this course once, so please make it as beneficial as possible for your future. The syllabus has been designed for breadth and depth of relevance across a large number of fields (as this book amply illustrates). Hard work put into the course will pay off in a stronger quantitative skill set for whatever your future studies will be.

Advice to Instructors

The meaning of each notation in Table 0.0.1 should be discussed when it first naturally arises in class (typically within the first 2 weeks). Students should be reminded about looking at Table 0.0.1 at the start of class for the first 6 or 7 meetings of the course (since new students are entering the course each day in the early weeks, and the notation in Table 0.0.1 is unfamiliar to nearly all students).

Some of the order of development of the linear algebra material in this book *differs from how textbooks on linear algebra are traditionally written*, and an instructor should be aware of these aspects before teaching the course (to avoid tacitly invoking ideas or results before they have actually appeared). This is motivated by two considerations: giving concrete definitions and numerical experience before more conceptual perspectives (without diminishing the importance of the latter), and recognizing that workers in other scientific fields always use software for certain computations (and we don't want to discuss procedures that most students will never encounter again if there are other ways to teach the essential concepts). Here are the main points:

(i) The notion of "linear subspace" (of $\mathbf{R}^n$) is initially defined in this book as the span of a finite collection of vectors. The equivalence with the more conceptual definition via closure under linear combinations is given only much later, in Section 21.3 (see Theorem 21.3.7). So in particular, the fact that the solutions to a homogeneous system of linear equations in n unknowns is a linear subspace of $\mathbf{R}^n$ is not essentially a tautology (in effect, it requires something akin to the fact that a vector subspace of a finite-dimensional vector space is finitely generated); explicit examples in Chapters 4 and 5 indicate to the student early on that this fact should be true.

(ii) Since "linear subspace" is defined in terms of span, we initially define "dimension" of a linear subspace V of $\mathbf{R}^n$ (in Chapter 4) to be the smallest size of a spanning set of V, so there is no well-definedness issue with the concept of dimension. We later define a "basis" of a linear subspace V of $\mathbf{R}^n$ in Chapter 5 to be a spanning set of V with the smallest size, so by definition all bases of V have size $\dim V$. Computing the dimension when it is at most 3 is discussed in Chapter 5, and some facts about dimension are stated and used long before they can be logically proved because *linear independence is postponed to Chapter* 19 (delaying this concept until it is really needed reduces cognitive overload among a large number of early definitions).

This approach shifts the burden of work to computing dimension and showing that a basis is the same thing as a linearly independent spanning set. This latter fact is recorded as Theorem 19.2.3, the proof of which is given in Appendix B.1. We use orthogonality and the Gram–Schmidt process as the foundation for our approach to the logical development of linear independence and dimension. This has the merit of being rather visual right from the start, and by the end of the course the student is fully aware of all of the usual ways of thinking about the core concepts of linear algebra developed in the course over $\mathbf{R}$. Many linear algebra proofs are referenced to Appendix B (of course, there is no circular reasoning).

(iii) The mechanics of matrix multiplication are initially explained in Section 14.3 via a method that displays the information in a more visually convenient way than literally writing one matrix to the left of another. This is mentioned on the Wikipedia page for matrix multiplication and deserves to be more widely known.

(iv) The Gram–Schmidt process as presented in Chapter 19 (see in particular Section 19.2 for the algorithm, and Section 19.3 for many worked examples) does *not* normalize the output vectors to be unit vectors. The rationale for this is explained in Remark 19.3.3.

(v) Gaussian elimination is *not* developed at all in this book (though bare-hands work with 2 equations in 3 unknowns is included, and LU-decompositions are introduced and used in Part IV).

(vi) The emphasis on eigenvalues and eigenvectors is on what they *mean* (algebraically and geometrically) and how they are *used* in all dimensions. We introduce the characteristic polynomial and computation of eigenvalues and eigenvectors only for the 2-dimensional case, but students learn *in all dimensions* how to check that an eigenvector given "out of thin air" really is an eigenvector and how to compute its eigenvalue. (Appendix E discusses characteristic polynomials in general.)

(vii) The notion of 2×2 determinants is introduced (in Section 18.2) but general $n \times n$ determinants are *not* discussed in the main text (though developed in Appendix E to provide a basic introduction for interested students to read on their own); the reason is that the course doesn't have compelling applications of general determinants (see (iii) above). In particular, the discussion of *computing* matrix inverses is founded upon LU- and QR-decompositions; this is how inverses are really computed outside pure-math settings (beyond the 2×2 case and other special situations).

(viii) We do *not* use complex numbers. Experience shows that complex numbers create too much unease and/or confusion for many students, so that aspect of linear algebra is best postponed to a later course where it serves a compelling purpose (e.g., linear ODE's or quantum mechanics). Much of the discussion of eigenvalues focuses on the symmetric case (such as for Hessians), where the Spectral Theorem has a starring role, but we also treat some non-symmetric examples.

Acknowledgments. This book is the outcome of a 4-year collaboration of many people in the Stanford Math department, at all levels: professors, lecturers, postdocs, and graduate students. The Math department is grateful to colleagues from other departments at Stanford and at other institutions for their feedback and/or suggestions at various stages of the process: Kyle Bagwell (Economics), Dan Boneh (CS), Katie Bouman (Caltech, Computational Imaging), Pat Burchat (Physics), Chris Chidsey (Chemistry), Marlene Cohen (University of Pittsburgh, Neuroscience), Keith Conrad (University of Connecticut, Math), John Duchi (Statistics & EE), Marcus Feldman (Biology), Iain Johnstone (Statistics & Biostatistics), Shamit Kachru (Physics), Roger Lee (University of Chicago, Financial Math), Chris Makler (Economics), Ian Morrison (Fordham University, Math), Jonathan Pritchard (Biology), Noah Rosenberg (Biology), Omar Rutledge (Neurology), Mehran Sahami (CS), Jonathan Taylor (Statistics), Carl Wieman (Physics). We are also grateful to many Math 51 students who have provided feedback to improve the book.

Part I

Geometry of vectors and projections

"And I cherish more than anything else the Analogies, my most trustworthy masters. They know all the secrets of Nature . . . "

J. Kepler

". . . the geometry of n dimensions has no mathematical characteristics essentially different from those of ordinary geometry. Spaces of $4, 5, \ldots$ dimensions, as we have defined them, exist for the mathematician precisely in the same way as space of 3 dimensions exists; and they may be studied by the same methods."

C. Segre [Seg, p. 463]

Overview of Part I

This Part, which comprises Chapters 1–7, is primarily about vectors and how to do geometry with them. Using geometric experience in 2 and 3 dimensions as the take-off point (e.g., Pythagorean Theorem and the Law of Cosines from plane geometry, and daily life in a 3-dimensional world), we develop a powerful geometric and algebraic *language* (called linear algebra) for working with the collection $\mathbf{R}^n$ of all ordered n-tuples of numbers $(x_1, \ldots, x_n)$ (also called n-*vectors*) for any $n \geq 1$. Examples are given to demonstrate the genuine utility of $\mathbf{R}^n$ for $n > 3$ in the study of real-world questions (stemming from the fact that many mechanical systems, data sets, networks, and biological and economic models often involve the interdependence of a large number of ingredients).

Notions such as distance, angle, and (linear) subspace are defined for $\mathbf{R}^n$ extending the notions of distance, angle, and plane that are more readily visualized in our 3-dimensional experience. In particular, the concept of *dimension* (of a subspace) is introduced, providing a precise way to develop "geometric intuition" while navigating around inside $\mathbf{R}^n$ with $n > 3$. With a bit of practice, one can eventually "feel" that certain results in high dimensions ought to be true, even though strictly speaking our visual experience is limited to dimension at most 3; this illustrates the power of mathematical concepts to provide us with novel yet sure-footed ways to gain insight into situations where it may have initially seemed impossible to argue in a geometric manner.

Reasoning in higher dimensions turns out not to be at all mysterious, and it is even liberating by enlarging the array of situations where the human brain's capacity for visual thought is a good guide. Along the way in this first Part of the book we discuss two concrete applications of the initial circle of ideas we develop in vector geometry: to linear regression (finding the best-fit line to a set of data points by using the geometric perspective of linear algebra that will later guide us into more sophisticated viewpoints on curve-fitting) and to statistics (with correlation). A wide array of applications of higher-dimensional considerations arises throughout the rest of the book.

A student who enters this course with an eye towards the differential calculus material (which is aimed at tackling optimization problems for functions of many variables, including appropriate analogues of single-variable calculus notions such as: derivative, Chain Rule, first-derivative test, and second-derivative test) may now be wondering: why is it appropriate to devote this entire first Part of the book to vector geometry before embarking on multivariable calculus aspects?

The answer is seen by thinking back to the experience of learning single-variable calculus. In that subject, the notion of derivative encodes the tangent line as the "best linear approximation" at a point on the graph of a function. Thus, a real understanding of single-variable derivatives (and geometric intuition for their properties) requires first understanding graphs of straight lines both algebraically and geometrically. Linear algebra is the multivariable replacement for "graphs of straight lines", and vector geometry is the foundation for visualization in linear algebra. Thus, a good understanding of multivariable differential calculus requires first learning some ideas and computational techniques in linear algebra. The interplay between linear algebra (in its geometric and algebraic aspects) and multivariable differential calculus is a recurring theme in this book.

1. Vectors, vector addition, and scalar multiplication

The goal of this chapter is to introduce vectors and some ways to manipulate them. By the end of this chapter, you should be able to:

- add vectors, and multiply vectors by scalars;
- meaningfully interpret linear combinations of vectors arising in science and data analysis;
- compute the length (also called magnitude) of a vector.

1.1. Vectors (and scalars).

Example 1.1.1. You are flying a drone in the sky near your house. How do you describe exactly where it is? You could mark the spot on the ground directly below the drone and measure (say in units of feet):

- how far *north* of your house that spot on the ground is;
- how far *east* of your house that spot on the ground is;
- how *high* in the sky the drone is above that spot on the ground.

Thus the position of the drone is completely specified by these three numbers:

$$(\text{distance north of house, distance east of house, height}).$$

If the position is $(200, 300, 25)$ it means that the spot on the ground is 200 feet north and 300 feet east of your house, and the drone is 25 feet in the air above it. If the spot is instead 200 feet *south* of your house (and keep the same distances east and in the air) then the position is $(-200, 300, 25)$. ■

What we have just described is called the *displacement vector* from the house to the drone. It is an example of a "3-vector":

Definition 1.1.2. For a whole number n, an *n-vector* is a list of n real numbers. We denote by $\mathbf{R}^n$ the collection of all possible n-vectors.

Example 1.1.3. For instance, $(200, 300, 25)$ is a 3-vector and $(-3, 4.9, \frac{1}{2}, 0, 1)$ is a 5-vector. We will usually write vectors "top to bottom", so the 3-vector just mentioned is written as $\begin{bmatrix} 200 \\ 300 \\ 25 \end{bmatrix}$. ■

The reason to prefer writing vectors vertically rather than horizontally is because of certain features of "matrix algebra" to be used a lot later on in this book.

Definition 1.1.4. To distinguish ordinary real numbers from vectors, we use the word *scalar* to refer to a real number.

Example 1.1.5. For instance, 1.3289 is a scalar but $\begin{bmatrix} 0.2 \\ -0.9 \end{bmatrix}$ is a vector. ■

As in high school algebra, symbols such as $x, y, z, a, b, c, \ldots$ denote numbers (or scalars, as we shall often say). In linear algebra, bold-face symbols $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{v}, \mathbf{w}, \mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$ denote vectors. Some books denote vectors as $\vec{x}, \vec{y}, \vec{z}, \vec{v}, \vec{w}, \vec{a}, \vec{b}, \vec{c}, \ldots$, and to distinguish vectors from scalars in handwriting you may want to use such arrow notation (or "half-arrow" variants $\overrightharpoon{x}, \overrightharpoon{y}, \overrightharpoon{z}, \overrightharpoon{v}, \overrightharpoon{w}, \overrightharpoon{a}, \overrightharpoon{b}, \overrightharpoon{c}, \ldots$ to save time).[1]

The aim of a large part of this book is to teach you *how to do both algebra and geometry with vectors* (and why it is so useful). This subject is called linear algebra. We will use algebra with vectors to develop the derivative aspects of multivariable calculus. In Math 52 and Math 53 these ideas are used to go further, developing multivariable integration and the study of ordinary differential equations. (If you later wish to supplement this with additional linear algebra, good options are Math 104 or Math 113 depending on your mathematical preferences; see Section 26.5.)

---

[1] In the early 1900's, several attempts to achieve universal notation for vectors among all scientists ended in total failure.

For $n = 2$ or $n = 3$, we can visualize n-vectors as arrows. For example, with $n = 2$, we can visualize the vector $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ as an arrow in the coordinate plane $\mathbf{R}^2$ pointing *from* the origin $(0, 0)$ *to* the point (v_1, v_2), and similarly for a 3-vector by working in $\mathbf{R}^3$. Examples are shown in Figure 1.1.1 below.

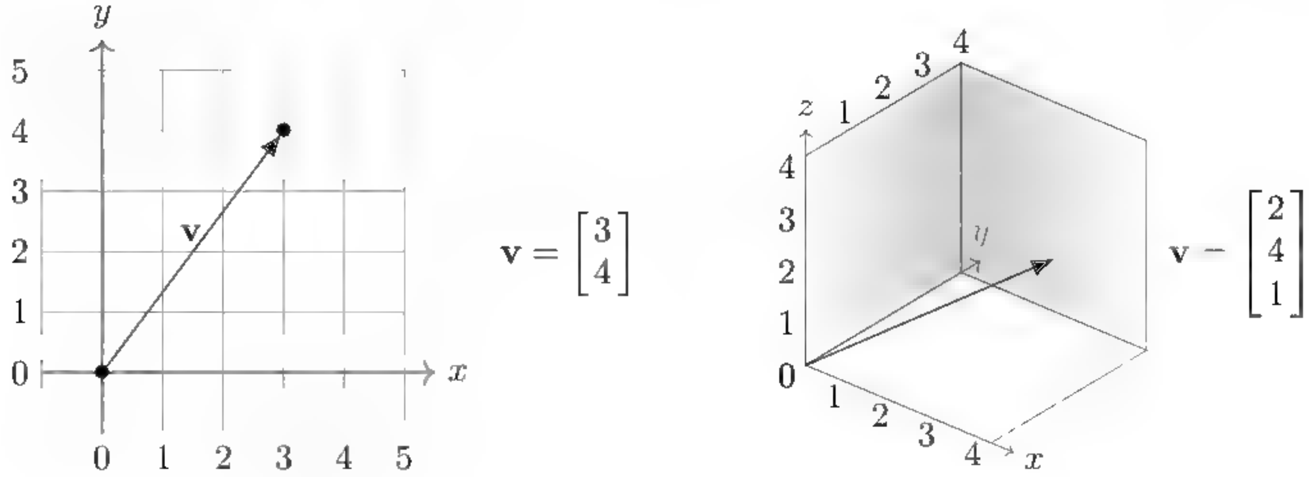


FIGURE 1.1.1. Visualization of a 2-vector and a 3-vector

1.2. Vectors in physical science, social science, and data science. The words *vector* and *vehicle* have the same Latin root: vehere ("to carry"). Can you see why? Here are two sources of vectors:

(V1) In the physical sciences, vectors are used to represent quantities that have both a *magnitude* and a *direction*. Examples of such quantities include displacement, velocity, force, and angular momentum. For instance, the vector in Example 1.1.1 is the displacement vector from the house to the drone. In general the *displacement vector* from one point P to another point Q is the vector represented by the arrow connecting those two points with the arrowhead at Q.

In a lot of physical examples, n is at most 3. However, in physical systems having many parts, it is convenient to group together various 3-vectors from the various parts into a single N-vector with $N > 3$. Thus, it is *incorrect* to think that linear algebra in $\mathbf{R}^n$ for $n > 3$ is not relevant to the study of the natural world! Many scientific phenomena (protein folding, the behavior of dynamical systems, the spread of a disease through a population, the balancing of forces in a mechanical system consisting of many parts, etc.) depend on a rather large number of unknowns, and linear algebra in $\mathbf{R}^n$ for large n is *essential* in their study.

The mathematical models used to study these phenomena typically involve collecting such data into vectors in $\mathbf{R}^n$ for various n. For example, genetic work often uses $\mathbf{R}^n$ for $n \approx 200{,}000$ (see Example 1.2.3). **The utility of $\mathbf{R}^n$ for $n > 3$ pervades many scientific disciplines, far beyond data science. The "magnitude and direction" viewpoint is NOT a meaningful definition of "n-vector" when $n > 3$.**

The importance of n-vectors with $n > 3$ was foreseen by the great physicist J.C. Maxwell who said in 1870 about the then-new theory of 3-vectors [Max, pp. 215-229] that

> *"... [it is] a branch of mathematics which, when it shall have been thoroughly understood ..., will become, perhaps under some new name, a most powerful method of communicating truly scientific knowledge ... "*

(V2) In data science, economics, and many industrial applications of mathematics, vectors are used to keep track of collections of numerical data. This type of example is much more varied than the examples arising from natural sciences, and nearly always n is *very large*.

Example 1.2.1. (Climate data) To keep track of the daily variation of temperature at Stanford in a given year (ignoring leap years), we can arrange it into a 365-vector

$$\mathbf{T} = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_{365} \end{bmatrix}$$

where T_1 is the noontime temperature on day 1, T_2 is the noontime temperature on day 2, and so forth.

To record more precise temperature information, say hourly rather than daily, you might measure the temperature at each hour on each day and correspondingly use an n-vector for $n = 24 \times 365$. ■

Example 1.2.2. (Grade data) Suppose that there are 100 students in a class. We can keep track of all of their grades on the final exam by forming a 100-vector

$$\mathbf{E} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_{100} \end{bmatrix},$$

where E_1 is the final exam grade of the first student, E_2 the final exam grade of the second student, and so on. Similarly, we can form vectors that represent their scores on the first and second midterms respectively:

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{100} \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_{100} \end{bmatrix}.$$

■

Example 1.2.3. (Genomic data) When trying to analyze genetic information, such as to predict the risk of disease, it is often helpful to encode part of a person's genome by an n-vector $\mathbf{G}$ with n very large, such as around 200,000, which encodes information only about certain positions in the human genome (called single nucleotide polymorphisms, or SNP's).

Thus, for example, the kth entry of the vector $\mathbf{G}$ gives information about the kth polymorphism. In the simplest possible version, one would enter 0 if the person's DNA at this spot coincides with the most frequent version found among the human population. Otherwise – that is, if the person's DNA *doesn't* match with the most frequent version found among the whole population – we would put a 1 in entry k of $\mathbf{G}$. (This is a simplification of what is usually done; for example one certainly wants to take account of both copies of a chromosome.) ■

Up to this point, calling these "vectors" doesn't help at all. They are just lists of numbers that give information about a situation, and we have only introduced some new language and notation; that is not progress. But we will soon see that linear algebra gives us powerful tools to organize and work with information encoded in such lists of numbers.

1.3. Operations on vectors and their visual meaning. We now introduce two algebraic operations permitted on vectors: *vector addition* and *scalar multiplication*.

Definition 1.3.1. The *sum* $\mathbf{v}+\mathbf{w}$ of two vectors is defined precisely when $\mathbf{v}$ and $\mathbf{w}$ are n-vectors for the same n. In that case, we define their sum by the rule $\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} v_1+w_1 \\ v_2+w_2 \\ \vdots \\ v_n+w_n \end{bmatrix}$.

Definition 1.3.2. We *multiply* a scalar c against an n-vector $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$ by the rule $c \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} cv_1 \\ cv_2 \\ \vdots \\ cv_n \end{bmatrix}$.

Example 1.3.3. (vector addition in $\mathbf{R}^2$ and $\mathbf{R}^3$)

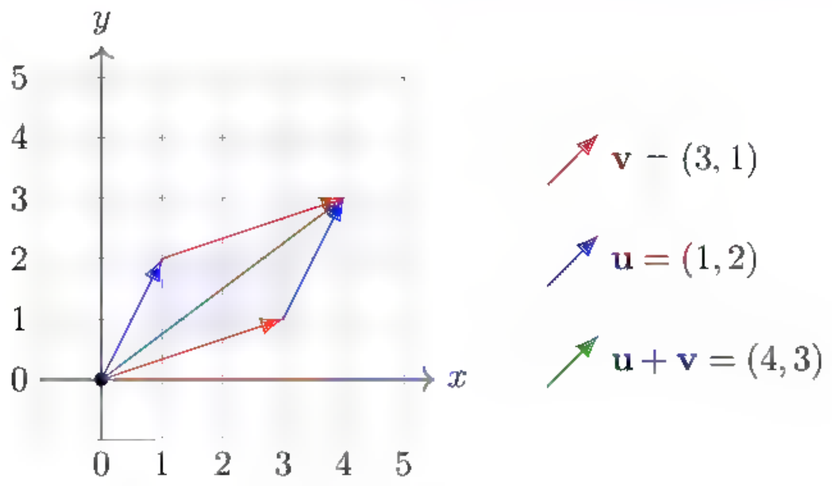


FIGURE 1.3.1. Addition of vectors in $\mathbf{R}^2$

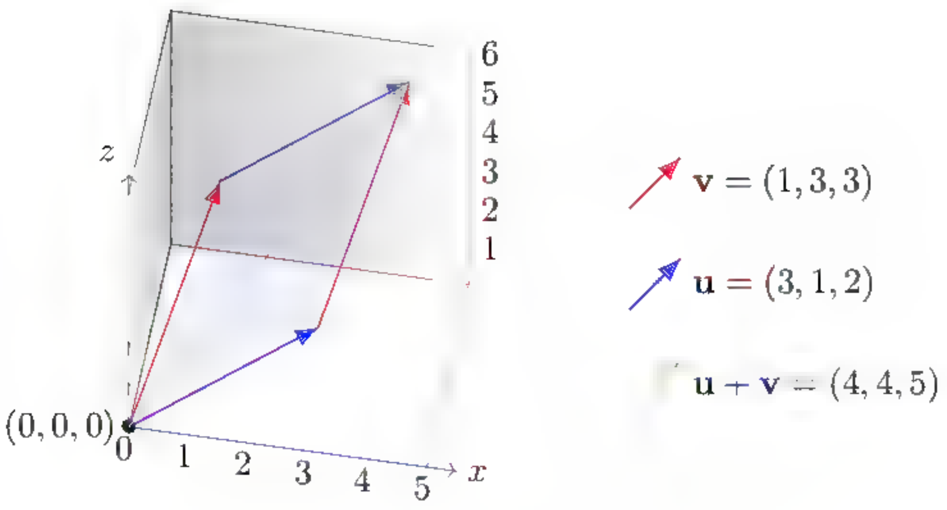


FIGURE 1.3.2. Addition of vectors in $\mathbf{R}^3$

As in Figure 1.3.1 above for $n = 2$ and Figure 1.3.2 above for $n = 3$, the vector $\mathbf{u}+\mathbf{v}$ is represented by the diagonal arrow in the parallelogram with one vertex at the origin and two edges given by $\mathbf{u}$ and $\mathbf{v}$. This description of vector addition for $n = 2$ and $n = 3$ is called the *parallelogram law*.

■

We define subtraction $\mathbf{v} - \mathbf{w}$ as we did addition, or equivalently $\mathbf{v} - \mathbf{w} = \mathbf{v} + (-1)\mathbf{w}$. It is very common to use both the operations at once – for example, we can form $2\mathbf{v} + 3\mathbf{w}$, where $\mathbf{v}$ and $\mathbf{w}$ are n-vectors. Here is the relevant terminology:

Definition 1.3.4. A *linear combination* of two n-vectors $\mathbf{v}, \mathbf{w}$ is an n-vector $a\mathbf{v} + b\mathbf{w}$ for scalars a, b. More generally, a linear combination of k such n-vectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$ is $a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \cdots + a_k\mathbf{v}_k$ for scalars $a_1, a_2, \dots, a_k$. (In physical sciences, this is often called a "superposition" of $\mathbf{v}_1, \dots, \mathbf{v}_k$.)

Example 1.3.5. Here are some numerical examples of the above concepts.

(i) $\begin{bmatrix} 2 \\ 1 \\ -8 \end{bmatrix} + \begin{bmatrix} 3 \\ 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 2+3 \\ 1+2 \\ -8+4 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \\ -4 \end{bmatrix}$

(ii) $5 \cdot \begin{bmatrix} -1 \\ 8 \end{bmatrix} = \begin{bmatrix} 5 \cdot (-1) \\ 5 \cdot 8 \end{bmatrix} = \begin{bmatrix} -5 \\ 40 \end{bmatrix}$

(iii) $\begin{bmatrix} 2 \\ 1 \\ 8 \end{bmatrix} - \begin{bmatrix} 3 \\ 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 2-3 \\ 1-2 \\ 8-4 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 4 \end{bmatrix}$

(iv) $3 \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + 6 \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \cdot 1 + 6 \cdot 1 \\ 3 \cdot 2 + 6 \cdot (-1) \\ 3 \cdot 3 + 6 \cdot 2 \end{bmatrix} = \begin{bmatrix} 9 \\ 0 \\ 21 \end{bmatrix}$

(v) $2 \begin{bmatrix} -1 \\ 0 \\ 1 \\ 2 \end{bmatrix} + 3 \begin{bmatrix} 2 \\ 1 \\ -3 \\ 4 \end{bmatrix} - 5 \begin{bmatrix} 0 \\ 2 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \cdot (-1) + 3 \cdot 2 - 5 \cdot 0 \\ 2 \cdot 0 + 3 \cdot 1 - 5 \cdot 2 \\ 2 \cdot 1 + 3 \cdot (-3) - 5 \cdot (-2) \\ 2 \cdot 2 + 3 \cdot 4 - 5 \cdot 1 \end{bmatrix} = \begin{bmatrix} 4 \\ -7 \\ 3 \\ 11 \end{bmatrix}$

■

Example 1.3.6. The expression $\begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 \\ 4 \end{bmatrix}$ *makes no sense*: one can't "add" a 3-vector and a 2-vector. ■

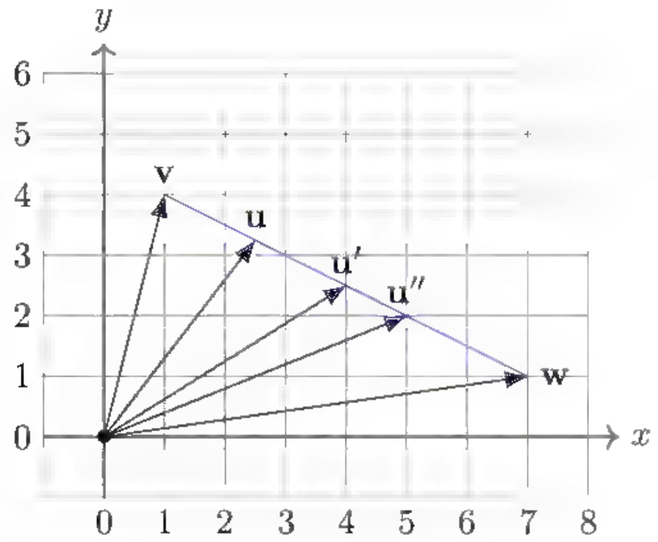
Remark 1.3.7 (online resource). The video series "Essence of Linear Algebra" at 3blue1brown provides illuminating visualizations of a variety of concepts in linear algebra. Its first video relates to the material in this chapter.

Alert. Please be aware that many online tutorials/references deviate from how things are done in this book: they may develop material in a different order, rely on concepts or techniques that we do not use, and/or formulate important definitions or results in a more abstract or less useful manner. Two guiding principles in the writing of this book were to give (i) concrete definitions and numerical experience before more conceptual perspectives (without diminishing the importance of the latter), (ii) concepts, techniques, and results that are applicable uniformly to $\mathbf{R}^n$ for all n (important for many contemporary applications of linear algebra and multivariable optimization across many fields).

By the end of the course you will know the standard main concepts and results (and beyond!), but the route to get there is not the traditional one. The approach here is designed to provide a more confident and applicable command of the material than the traditional order in which it is taught.

Example 1.3.8. A special type of linear combination that arises in applications such as linear programming, weighted averages, and probability theory is *convex combination*: in the case of two n-vectors $\mathbf{v}$

and $\mathbf{w}$ this means a linear combination of the form $(1-t)\mathbf{v}+t\mathbf{w}=\mathbf{v}+t(\mathbf{w}-\mathbf{v})$ with $0 \le t \le 1$. This adds to $\mathbf{v}$ a portion (given by t) of the displacement $\mathbf{w}-\mathbf{v}$ from $\mathbf{v}$ to $\mathbf{w}$. It has the geometric interpretation (for $n=2,3$) of being a point on the line segment *between* the tips of $\mathbf{v}$ and $\mathbf{w}$; e.g., it is $\mathbf{v}$ when $t=0$, it is the midpoint when $t=1/2$, and it is $\mathbf{w}$ when $t=1$. Figure 1.3.3 below illustrates this in $\mathbf{R}^2$ with $t=1/4, 1/2, 2/3$ corresponding to the indicated vectors $\mathbf{u}, \mathbf{u}', \mathbf{u}''$[2] respectively.



$\mathbf{v}=(1,4)$
$\mathbf{w}=(7,1)$
$\mathbf{u}=(3/4)\mathbf{v}+(1/4)\mathbf{w}=(5/2,13/4)$
$\mathbf{u}'=(1/2)\mathbf{v}+(1/2)\mathbf{w}=(4,5/2)$
$\mathbf{u}''=(1/3)\mathbf{v}+(2/3)\mathbf{w}=(5,2)$

FIGURE 1.3.3. Blue segment of vectors $(1-t)\mathbf{v}+t\mathbf{w}=\mathbf{v}+t(\mathbf{w}-\mathbf{v})$ for $0 \le t \le 1$

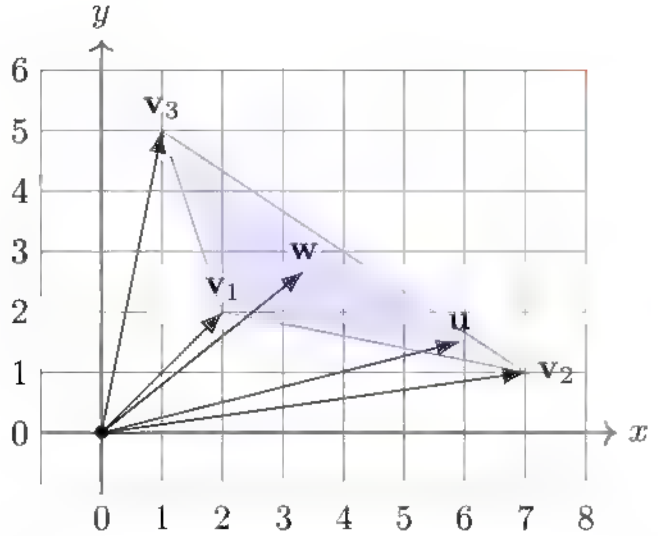
For any n-vectors $\mathbf{v}_1,\dots,\mathbf{v}_k$, a *convex combination* of them means a linear combination

$$t_1\mathbf{v}_1+\cdots+t_k\mathbf{v}_k$$

for which all $t_j \ge 0$ and the sum of the coefficients is equal to 1; that is, $t_1+\cdots+t_k=1$. When the k coefficients are all equal, which is to say every t_j is equal to $1/k$, this is the *average* (sometimes called the *centroid*) of the k vectors. Working in $\mathbf{R}^2$, a convex combination such as

$$\frac{2}{3}\mathbf{v}_1+\frac{1}{4}\mathbf{v}_2+\frac{1}{12}\mathbf{v}_3$$

is a point inside the polygon with vertices given by the $\mathbf{v}_j$'s, with its distance to each $\mathbf{v}_j$ weighted by the corresponding coefficient. If all coefficients are equal, it is the "center of mass" of the polygon. Two convex combinations are shown in Figure 1.3.4 below (see $\mathbf{u}$ and $\mathbf{w}$ there).



$\mathbf{v}_1=(2,2)$
$\mathbf{v}_2=(7,1)$
$\mathbf{v}_3=(1,5)$
$\mathbf{w}=(1/3)\mathbf{v}_1+(1/3)\mathbf{v}_2+(1/3)\mathbf{v}_3$
$=(10/3,8/3)$
$\mathbf{u}=(1/10)\mathbf{v}_1+(4/5)\mathbf{v}_2+(1/10)\mathbf{v}_3$
$=(59/10,3/2)$

FIGURE 1.3.4. The vector $\mathbf{w}$ is the "center of mass" of $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$.

■

---

[2] Notation such as $\mathbf{u}'$ and $\mathbf{u}''$ does *not* denote derivatives (indeed, it would make no sense). In math, such dashes denote additional versions of some common structure (such as vectors here, or t, t', t'' for time variables, etc.) when not functions.

Example 1.3.9. In Figure 1.3.3 we described the line segment with specified endpoints $\mathbf{v} = (1,4)$ and $\mathbf{w} = (7,1)$: it consists of points $t\mathbf{v} + (1-t)\mathbf{w} = \mathbf{v} + t(\mathbf{w} - \mathbf{v})$ for a "parameter" t that varies from 0 to 1. But we can allow t to vary through the entirety of $\mathbf{R}$ (not just the interval $[0,1]$) and thereby get the entire line in $\mathbf{R}^2$ through those two points. To be specific, the linear combinations

$$(1-t)\mathbf{v} + t\mathbf{w} = \mathbf{v} + t(\mathbf{w} - \mathbf{v}) = \begin{bmatrix} 1 \\ 4 \end{bmatrix} + t\begin{bmatrix} 6 \\ -3 \end{bmatrix} = \begin{bmatrix} 1+6t \\ 4-3t \end{bmatrix}$$

for all t (no constraint) are the points of the line ℓ in $\mathbf{R}^2$ through $(1,4)$ and $(7,1)$.

The slope of the line ℓ is $(4-1)/(1-7) = 3/(-6) = -1/2$, and it passes through $(1,4)$, so the "point-slope" equation of ℓ is $y - 4 = -(1/2)(x-1) = -x/2 + 1/2$, or equivalently $y = -x/2 + 9/2$. We have shown that ℓ admits the "parametric form" $(1+6t, 4-3t)$ with t varying through $\mathbf{R}$. In Section 3.3 we will discuss the "parametric form" of a general line, first in $\mathbf{R}^3$ and then in $\mathbf{R}^n$ for any n. ■

Example 1.3.10. For an n-vector $\mathbf{x} = (x_1, \dots, x_n)$, the numbers $t_j(\mathbf{x}) = e^{x_j}/(e^{x_1} + e^{x_2} + \cdots + e^{x_n})$ are positive and sum to 1! Such expressions first arose in statistical mechanics. Convex combinations with coefficients of the form $t_1(\mathbf{x}), \dots, t_n(\mathbf{x})$ are used all over the place in machine learning (where the function sending $\mathbf{x}$ to $(t_1(\mathbf{x}), \dots, t_n(\mathbf{x}))$ is called the (unit) softmax function). ■

Example 1.3.11. Iterating the parameterization of a line segment gives rise to an important construction in computer graphics called *Bézier curves*. More specifically, if we define $L_{P_0,P_1}(t) = (1-t)P_0 + tP_1$ for two points P_0 and P_1 and $0 \le t \le 1$, then the quadratic curve

$$L_{P_0,P_1,P_2}(t) = L_{L_{P_0,P_1}(t),L_{P_1,P_2}(t)}(t) = (1-t)L_{P_0,P_1}(t) + tL_{P_1,P_2}(t) = (1-t)^2P_0 + 2(1-t)tP_1 + t^2P_2$$

for $0 \le t \le 1$ is a path from P_0 (at $t = 0$) to P_2 (at $t = 1$) with "control point" P_1 (through which the path usually doesn't pass!). Likewise,

$$L_{P_0,P_1,P_2,P_3}(t) = (1-t)L_{P_0,P_1,P_2}(t) + tL_{P_1,P_2,P_3}(t) = (1-t)^3P_0 + 3(1-t)^2tP_1 + 3(1-t)t^2P_2 + t^3P_3$$

for $0 \le t \le 1$ is a cubic curve from P_0 to P_3 with "control points" P_1 and P_2. Moving the control points gives a flexible array of curves with specified endpoints. Joining such curves end to end using suitable control points underlies computer fonts for letters and many computer graphics applications (such as Adobe Illustrator and Javascript). ■

Let us now record a very ubiquitous bit of terminology that we have already implicitly invoked.

In linear algebra, the phrase *point in* $\mathbf{R}^n$ means **exactly the same thing** as "n-vector" (as well as "vector", when we don't need to specify n). The mental image for a given situation may suggest a preference between the words "point" and "vector", such as "displacement vector" or "closest point", but there is absolutely no difference in the meanings of these words in linear algebra.

You might imagine that a "point" is the tip of an arrow emanating from $\mathbf{0}$, or that a "vector" is a directed line segment with specified endpoints (which you may slide around parallel to itself to visualize addition as in Figure 1.3.2). But during this course you will see that *in the context of linear algebra* it is best to forget that and to regard "point" and "vector" as meaning exactly the same thing (this has *no effect* on the tremendous utility in other fields of the linear algebra we will teach you).

1.4. Interpreting the vector operations in examples. Here are some examples of the real-life meaning of the vector operations, with reference to the examples above. Please check for yourself that they work as claimed, to make sure you understand the definition and interpretations of the vector operations.

Example 1.4.1. (differences of displacement as relative displacement) If $\mathbf{v}$ expresses the displacement *from* position 1 to position 2, and $\mathbf{w}$ expresses the displacement *from* position 2 to position 3, then $\mathbf{v} + \mathbf{w}$ expresses the displacement *from* position 1 to position 3.

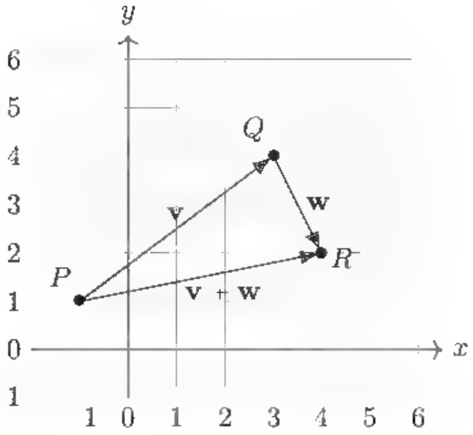


FIGURE 1.4.1. Net displacement as vector addition

For example, as in Figure 1.4.1, suppose a particle is moving in $\mathbf{R}^2$ beginning at the point $P = (-1, 1)$. The particle then moves to the point $Q = (3, 4)$. The *displacement* from P to Q is the vector

$$\mathbf{v} = \begin{bmatrix} 3 \\ 4 \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}.$$

This says that to get from P to Q the particle has to move 4 units to the right and 3 units up.

Now suppose the particle moves from Q to $R = (4, 2)$. The *displacement* from Q to R is the vector

$$\mathbf{w} = \begin{bmatrix} 4 \\ 2 \end{bmatrix} - \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

since to get from Q to R the particle has to move 1 unit to the right and 2 units *down*. We draw $\mathbf{w}$ in Figure 1.4.1 emanating from the tip of $\mathbf{v}$ to remind us of its physical significance as a displacement, as is also useful for visualizing $\mathbf{v} + \mathbf{w}$. The net displacement from P to R is

$$\begin{bmatrix} 4 \\ 2 \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \end{bmatrix},$$

and this agrees with the *sum* of the two intermediate displacements, as we can see in two ways:

$$\mathbf{w} + \mathbf{v} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} + \begin{bmatrix} 4 \\ 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}, \quad \text{or} \quad \mathbf{w} + \mathbf{v} = \left(\begin{bmatrix} 4 \\ 2 \end{bmatrix} - \begin{bmatrix} 3 \\ 4 \end{bmatrix}\right) + \left(\begin{bmatrix} 3 \\ 4 \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 4 \\ 2 \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}.$$

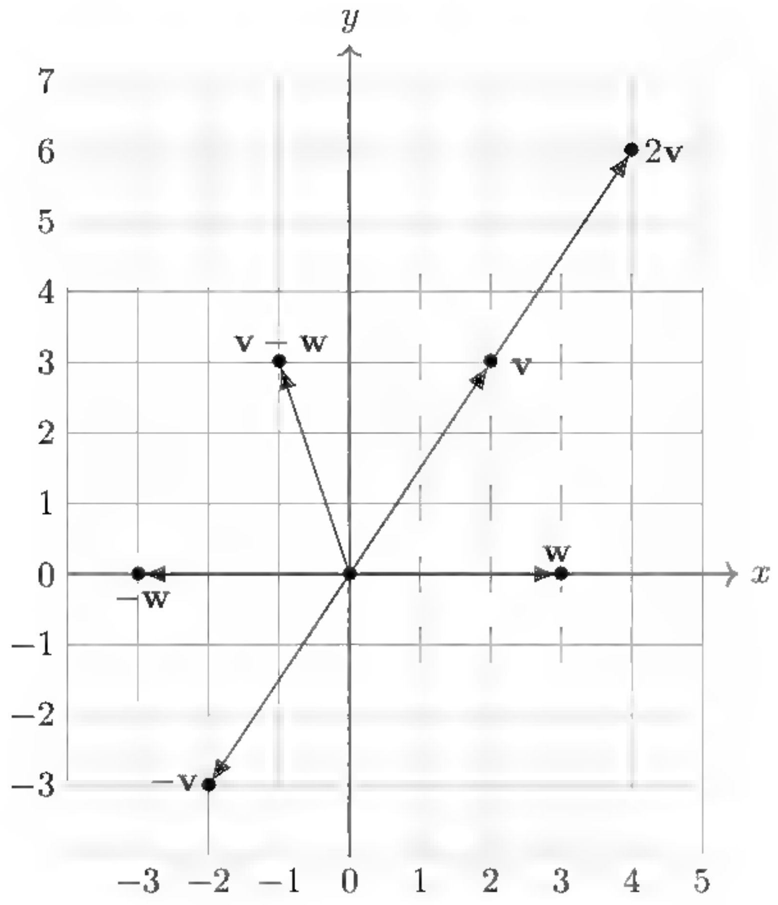
This shows why adding displacement vectors computes net displacement: $(\mathbf{u}_1 - \mathbf{u}_2) + (\mathbf{u}_2 - \mathbf{u}_3) = \mathbf{u}_1 - \mathbf{u}_3$.

■

Example 1.4.2. (velocities) If $\mathbf{v}$ describes the velocity of car A and $\mathbf{w}$ describes the velocity of car B then:

(i) $2\mathbf{v}$ describes the velocity of a car traveling in the same direction as A but moving twice as fast.
(ii) $-\mathbf{v}$ describes the velocity of a car that is moving as fast as A but in the opposite direction.
(iii) $\mathbf{v} - \mathbf{w}$ represents the velocity of car A *relative to* (i.e., viewed from) B.

Examples of this are shown in Figure 1.4.2 below.



$\mathbf{v} = (2, 3)$

$\mathbf{w} = (3, 0)$

$2\mathbf{v} = (4, 6)$

$\mathbf{v} - \mathbf{w} = (-1, 3)$

$-\mathbf{w} = (-3, 0)$

$-\mathbf{v} = (-2, -3)$

FIGURE 1.4.2. Some velocity vectors

Why is the order of subtraction $\mathbf{v} - \mathbf{w}$ in case (iii) rather than $\mathbf{w} - \mathbf{v}$? One way to understand this is to consider a special case: cars moving in the same direction on the same line, say car A at 90 km/hr and car B at 75 km/hr. The relative velocity of A from the viewpoint of B is then 15 km/hr, and $15 = 90 - 75$. Why is $75 - 90 = -15$ km/hr the reasonable "relative velocity" of B from the viewpoint of A?

Remark 1.4.3. In general, if we denote by $\mathbf{x}$ the "relative" velocity of car A from the viewpoint of someone in car B, it should mean that when we *add* the velocity $\mathbf{w}$ of car B to $\mathbf{x}$ then we get the velocity $\mathbf{v}$ of car A. Hence, $\mathbf{x} + \mathbf{w} = \mathbf{v}$, so $\mathbf{x} = \mathbf{v} - \mathbf{w}$. That explains the general case.

■

Example 1.4.4. (linear combination of grades) In Example 1.2.2, suppose that the total score for the class is weighted 25% for each of two midterms and 50% for the final exam. Arrange the grades on each exam in a list according to alphabetical order of the names, with respective grade vector $\mathbf{X}$ for the first midterm and $\mathbf{Y}$ for the second midterm, and final exam grade vector $\mathbf{E}$. These are all 100-vectors, and the vector

$$(0.25)\mathbf{X} + (0.25)\mathbf{Y} + (0.5)\mathbf{E} = \frac{1}{4}\mathbf{X} + \frac{1}{4}\mathbf{Y} + \frac{1}{2}\mathbf{E}$$

(which is a convex combination as in Example 1.3.8) is a 100-vector whose entries are the weighted scores for all students in the class (again, arranged in alphabetical order). ■

Example 1.4.5. (linear combination of temperature vectors) In Example 1.2.1, suppose that $\mathbf{T}_{2001}$, $\mathbf{T}_{2002}$, …, $\mathbf{T}_{2016}$ are 365-vectors that describe the daily average temperatures in Palo Alto (say in Celsius) in years 2001, 2002, …, 2016 (let's ignore February 29 in leap years). Then

$$\frac{1}{16}\left(\mathbf{T}_{2001} + \cdots + \mathbf{T}_{2016}\right)$$

is a 365-vector that tells us, for each given day, the *average* temperature in the years 2001–2016. For example, the first entry of this vector is the average January 1 temperature during this period. ■

Example 1.4.6. (quantum computation) In Google's advance [Aru1] on quantum computing, random quantum circuits were expressed as D-vectors for $D = 2^{53} \approx 10^{16}$. The computational process was modeled as a convex combination $F\mathbf{v} + (1 - F)\mathbf{w}$ [Aru2, IV.A, (2); IV.B, (24)] for D^2-vectors $\mathbf{v}$ and $\mathbf{w}$ respectively corresponding to an "ideal" output and "errors", and a "fidelity" scalar F in the range $10^{-3} \leq F \leq 10^{-2}$ [Aru2, XI.A, (112)]. We'll come back to this in Remark 20.1.11. ■

1.5. Legal and illegal operations of vector algebra.

All the familiar rules of usual algebra apply to the vector algebra operations we have seen so far (and are verified by just plugging in the algebraic formulas used in the definitions). For example,

$$c(\mathbf{v} + \mathbf{w}) = c\mathbf{v} + c\mathbf{w}$$

for any n-vectors $\mathbf{v}, \mathbf{w}$ and any scalar c. We give a more comprehensive list below in Theorem 1.5.2, but there's no point in memorizing that list since everything in it is reminiscent of a familiar property of addition and multiplication with numbers.

It is much more important at this stage to remember some things you *shouldn't* do.

- You are *never allowed* to add $\mathbf{v} + \mathbf{w}$ if $\mathbf{v}$ is an n-vector and $\mathbf{w}$ is an m-vector with $n \neq m$. For example, it is meaningful to carry out the computation
$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 4 \\ -5 \\ 6 \end{bmatrix} = \begin{bmatrix} 5 \\ -3 \\ 9 \end{bmatrix}$$
but the "sum" $\begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix}$ makes no sense. It also makes no sense to add a scalar to a vector.
- You cannot (and so should not try to) *multiply* or *divide* vectors to get another vector. In Chapter 2 we will define a special type of product for vectors called the "dot product." However, the dot product of two n-vectors is a *scalar*, not a vector.

Remark 1.5.1. In physics and engineering, a special "cross product" of 3-vectors (the output of which is also a 3-vector) shows up a lot. This has *no analogue* in $\mathbf{R}^n$ for $n \neq 3$, and it behaves very differently from products of numbers; e.g., it is neither commutative nor associative! If you're curious to learn about it, read the optional Appendix F after getting through Part I. **We do not use the cross product because we develop techniques applicable to $\mathbf{R}^n$ for all** n. The cross product is introduced and used in Math 52.

Theorem 1.5.2. For vectors in $\mathbf{R}^n$, we have:[3]

- (commutative law) $\mathbf{v} + \mathbf{w} = \mathbf{w} + \mathbf{v}$,
- (associative law) $(\mathbf{v} + \mathbf{v}') + \mathbf{v}'' = \mathbf{v} + (\mathbf{v}' + \mathbf{v}'')$,
- (distributive law) $c(\mathbf{v} + \mathbf{w}) = c\mathbf{v} + c\mathbf{w}$ for any scalar c,
- (more distributive and associative laws) $(c + c')\mathbf{v} = c\mathbf{v} + c'\mathbf{v}$ and $c(c'\mathbf{v}) = (cc')\mathbf{v}$ for scalars c, c'.

1.6. Length for vectors and distance in $\mathbf{R}^n$.

Returning to Example 1.1.1, how far is the drone from the house? The answer is given by a 3-dimensional version of the Pythagorean Theorem:

$$\text{distance} = \sqrt{(\text{distance north of house})^2 + (\text{distance east of house})^2 + (\text{height})^2}.$$

---

[3] Notation here such as $\mathbf{v}'$, $\mathbf{v}''$, c' does *not* denote derivatives (it would make no sense). In math, such dashes denote additional versions of some common structure (such as vectors, or scalars, or t, t', t'' for time variables, etc.) when not functions.

In other words, if the position of the drone is described by the vector $\mathbf{d} = (d_1, d_2, d_3)$, the distance of the drone from the house is given by $\sqrt{d_1^2 + d_2^2 + d_3^2}$.

This motivates a general definition (whose utility will become apparent with experience):

Definition 1.6.1. The *length* or *magnitude* of an n-vector $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$, denoted $\|\mathbf{v}\|$, is the number

$$\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2} \geq 0.$$

Note that the length is a scalar, and $\|-\mathbf{v}\| = \|\mathbf{v}\|$ (in accordance with the visualization of $-\mathbf{v}$ as "a copy of $\mathbf{v}$ pointing in the opposite direction") because signs disappear when squaring each $-v_j$.

If c is any scalar then $\|c\mathbf{v}\| = |c|\|\mathbf{v}\|$ (i.e., if we multiply a vector by c then the length scales by the factor $|c|$). For example, $(-5)\mathbf{v}$ has length $5\|\mathbf{v}\|$.

(In other references, you may see $\|\mathbf{v}\|$ called the "norm" of $\mathbf{v}$.)

Example 1.6.2. The length of the vector $\begin{bmatrix} 0.3 \\ -0.7 \\ 2.4 \end{bmatrix}$ is $\sqrt{(0.3)^2 + (-0.7)^2 + (2.4)^2} = \sqrt{6.34} \approx 2.52$. If we multiply this vector by 2 then we get the vector $\begin{bmatrix} 0.6 \\ -1.4 \\ 4.8 \end{bmatrix}$ whose length is $\sqrt{(0.6)^2 + (-1.4)^2 + (4.8)^2} = \sqrt{25.36} \approx 5.04 = 2(2.52)$. ■

Example 1.6.3. If a car is moving with velocity $\mathbf{v}$ then the magnitude $\|\mathbf{v}\|$ represents the speed of the car. By contrast, for the vector of daily temperatures in Example 1.2.1 the magnitude $\|\mathbf{T}\|$ does not have a physical meaning but it is still a measure of the "size" of $\mathbf{T}$. For instance, if we write down vectors $\mathbf{T}_{2015}$ and $\mathbf{T}_{2016}$ for daily temperatures at Stanford in 2015 and 2016 respectively and find that $\|\mathbf{T}_{2016}\|$ is bigger than $\|\mathbf{T}_{2015}\|$ then it suggests that on average 2016 was a warmer year than 2015 (since negative temperatures in Celsius essentially never occur in Palo Alto). ■

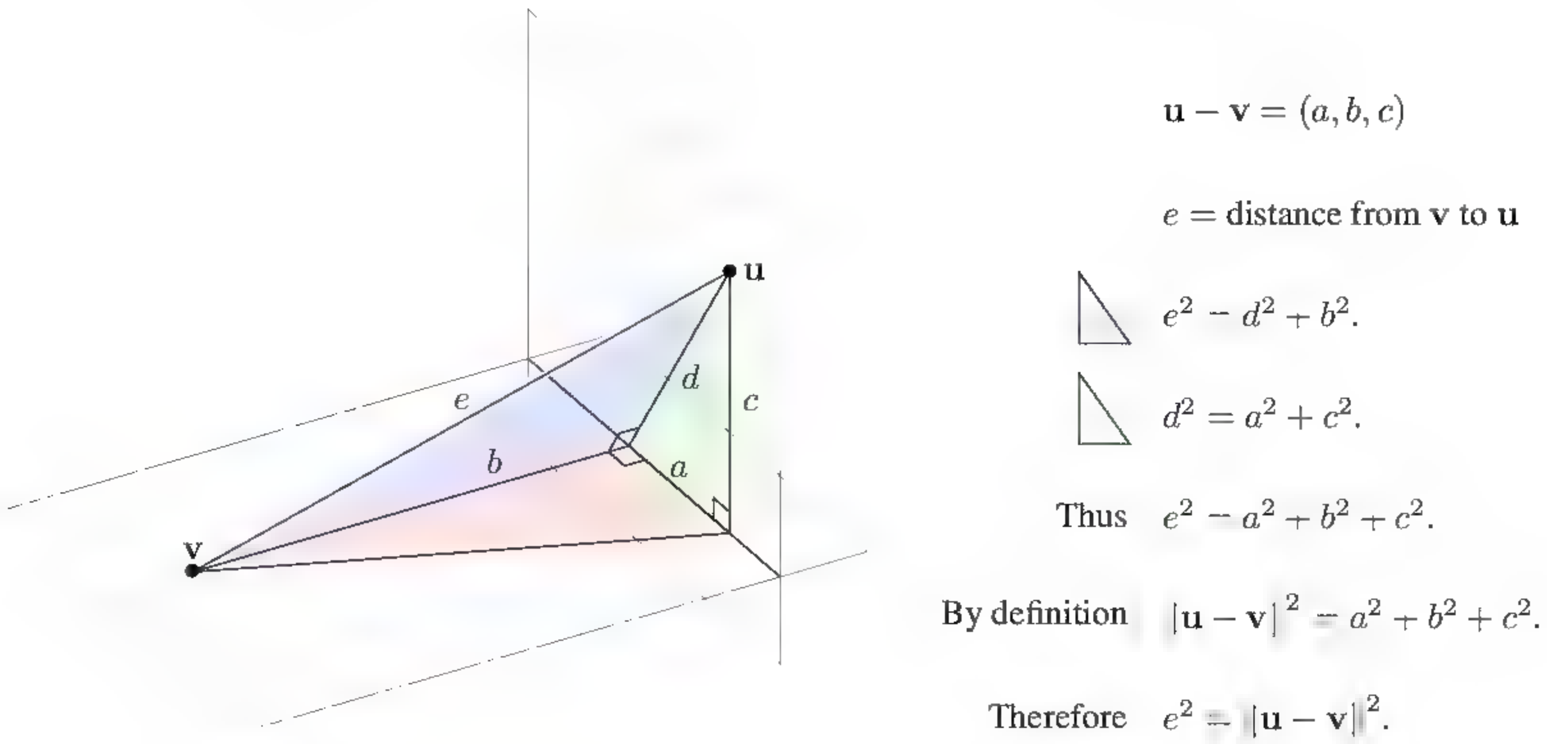
Definition 1.6.4. The *distance* between two n-vectors $\mathbf{x}, \mathbf{y}$ is defined to be $\|\mathbf{x} - \mathbf{y}\|$.

(This will soon be visualized as the familiar notion of distance between tips of arrows for $n = 2, 3$. In general it also equals $\|\mathbf{y} - \mathbf{x}\|$ since $\mathbf{y} - \mathbf{x} = -(\mathbf{x} - \mathbf{y})$ and any vector has the same length as its negative, so the order of subtraction doesn't matter.)

The preceding definition is just that: a definition. There is no "physical justification" to be given when $n > 3$. What is important is that (i) for $n = 2, 3$ we convince ourselves that it is the usual notion of distance, and (ii) for general n it satisfies reasonable *properties* for a notion of "distance" to provide helpful geometric insight. Such a property with general n will be discussed in Remark 1.6.12.

Let's convince ourselves that Definition 1.6.4 is the right concept when $n = 2, 3$. If $n = 2$, this follows from the Pythagorean Theorem in plane geometry (yielding the usual "distance formula" between points in $\mathbf{R}^2$) by thinking about displacement vectors: look at Figure 1.4.1, taking $\mathbf{x}$ to be $\mathbf{v} + \mathbf{w}$ there and $\mathbf{y}$ to be $\mathbf{v}$ there, so $\mathbf{x} - \mathbf{y} = \mathbf{w}$ is the displacement vector.

Example 1.6.5. Figure 1.6.1 below shows why Definition 1.6.4 is the usual notion of distance in $\mathbf{R}^3$, by computing the distance e between two vectors $\mathbf{u}$ and $\mathbf{v}$ via two applications of the 2-dimensional Pythagorean Theorem. The conclusion is that e equals $\|\mathbf{u}-\mathbf{v}\|$ as defined in Definition 1.6.4 for $n = 3$.



$\mathbf{u}-\mathbf{v} = (a, b, c)$

e = distance from $\mathbf{v}$ to $\mathbf{u}$

$e^2 = d^2 + b^2.$

$d^2 = a^2 + c^2.$

Thus $e^2 = a^2 + b^2 + c^2.$

By definition $\|\mathbf{u}-\mathbf{v}\|^2 = a^2 + b^2 + c^2.$

Therefore $e^2 = \|\mathbf{u}-\mathbf{v}\|^2.$

FIGURE 1.6.1. Pythagorean Theorem in $\mathbf{R}^3$

■

We next illustrate this using numerical examples. There is nothing special about the numbers we chose; the same considerations work quite generally in $\mathbf{R}^2$ and $\mathbf{R}^3$.

Example 1.6.6. Suppose that to drive from Pat's house to Casey's house, Pat had to drive 3 miles east, and then turn left and drive 1 mile north. Furthermore, suppose that to get to Sam's house from Pat's house, Pat had to drive 1 mile west and 2 miles south. How far would a bird fly, taking the most direct route, between Sam's house and Casey's house?

To figure this out, imagine a grid on the surface of the city where Pat, Sam, and Casey live. Set things up so that Pat's house is at the origin and the units on this grid are measured in miles: this is illustrated in Figure 1.6.2 below.

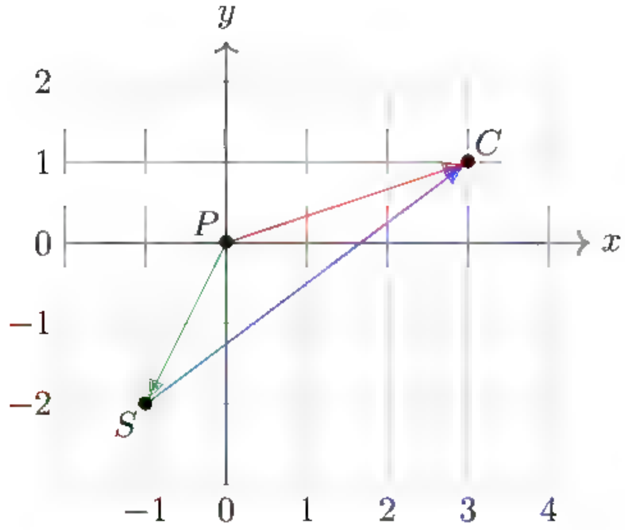


FIGURE 1.6.2. Houses of Pat, Casey, and Sam

Pat's house is represented by the vector $\begin{bmatrix}0\\0\end{bmatrix}$, Casey's house is represented by the vector $\begin{bmatrix}3\\1\end{bmatrix}$, and Sam's house is represented by the vector $\begin{bmatrix}-1\\-2\end{bmatrix}$. The distance the bird would fly is the distance from the vector $\begin{bmatrix}-1\\-2\end{bmatrix}$ to the vector $\begin{bmatrix}3\\1\end{bmatrix}$, which is the length of the difference:

$$\left\| \begin{bmatrix}3\\1\end{bmatrix} - \begin{bmatrix}-1\\-2\end{bmatrix} \right\| = \left\| \begin{bmatrix}4\\3\end{bmatrix} \right\| = \sqrt{4^2+3^2} = \sqrt{25} = 5.$$

So the bird would fly 5 miles to get from Sam's house to Casey's house.

The difference vector just used can be interpreted as an instance of the notion of *displacement vector* discussed in Example 1.4.1. ■

Example 1.6.7. For $\mathbf{v} = \begin{bmatrix}7\\3\end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix}4\\4\end{bmatrix}$, the distance between them is the length of the difference

$$\mathbf{v} - \mathbf{w} = \begin{bmatrix}7\\3\end{bmatrix} - \begin{bmatrix}4\\4\end{bmatrix} = \begin{bmatrix}3\\-1\end{bmatrix}.$$

This is shown in Figure 1.6.3 below.

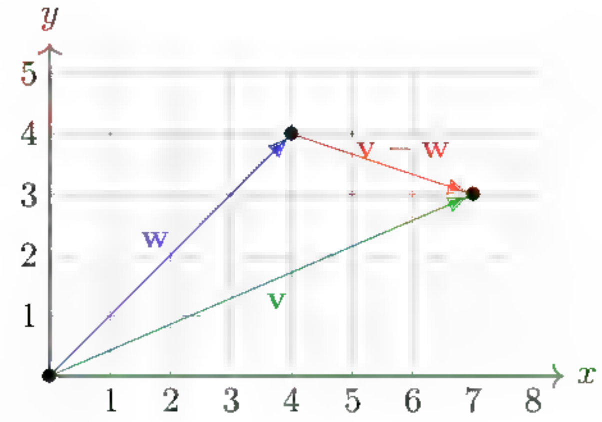


FIGURE 1.6.3. Distance as length of a difference vector

This length is $\sqrt{3^2+(-1)^2} = \sqrt{9+1} = \sqrt{10} \approx 3.1623$. (When we give decimal approximations in this book, it is solely for expository purposes; you are never expected to provide such approximations on exams, where *exact* numerical answers are always sufficient.) ■

Example 1.6.8. Let $\mathbf{u} = \begin{bmatrix}1\\1\end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix}3\\0\end{bmatrix}$. By Definition 1.6.4, the distance from $\mathbf{u}$ to $\mathbf{v}$ is

$$\|\mathbf{u} - \mathbf{v}\| = \left\| \begin{bmatrix}-2\\1\end{bmatrix} \right\| = \sqrt{4+1} = \sqrt{5}.$$

This is shown in Figure 1.6.4.

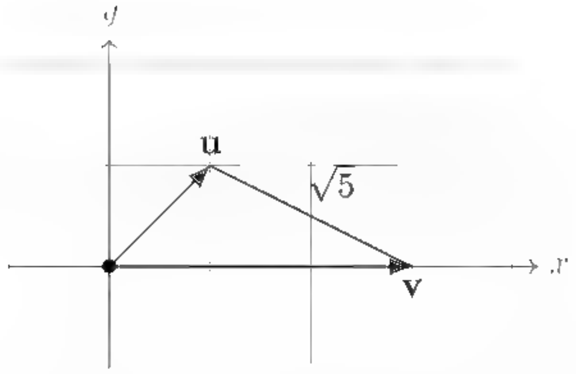


FIGURE 1.6.4. Two vectors with distance $\sqrt{5}$ from each other

■

Example 1.6.9. Let $\mathbf{u} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} 1/2 \\ \sqrt{3}/2 \end{bmatrix}$. What is the distance from $\mathbf{u}$ (with length 1) to $\mathbf{v}$ (with length 1)? By Figure 1.6.5 and trigonometry, $\mathbf{v}$ makes an angle of 60° with $\mathbf{u}$. Hence, the triangle is equilateral and so the distance is 1. Alternatively, we can compute with the difference vector:

$$\mathbf{u} - \mathbf{v} = \begin{bmatrix} 1/2 \\ -\sqrt{3}/2 \end{bmatrix}, \quad \text{so} \quad \|\mathbf{u} - \mathbf{v}\|^2 = \left(\frac{1}{2}\right)^2 + \left(\frac{-\sqrt{3}}{2}\right)^2 = \frac{1}{4} + \frac{3}{4} = 1 \text{ and hence } \|\mathbf{u} - \mathbf{v}\| = 1,$$

which again shows that the triangle formed by the unit-length edges $\mathbf{u}$ and $\mathbf{v}$ is indeed equilateral.

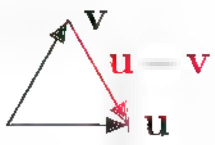


FIGURE 1.6.5. An equilateral triangle

■

Example 1.6.10. For[4] $\mathbf{v}' = \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix}$ and $\mathbf{w}' = \begin{bmatrix} 2 \\ 5 \\ 1 \end{bmatrix}$, the distance between them is the length of the difference

$$\mathbf{v}' - \mathbf{w}' = \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix} - \begin{bmatrix} 2 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -6 \\ 1 \end{bmatrix}.$$

This length is $\sqrt{1^2 + (-6)^2 + 1^2} = \sqrt{1 + 36 + 1} = \sqrt{38} \approx 6.1644$. ■

---

[4] Notation such as $\mathbf{v}'$ and $\mathbf{w}'$ here does *not* denote derivatives (it would make no sense). In math, such dashes denote additional versions of some common structure (such as vectors here, or t, t', t'' for time variables, etc.) when not functions.

Example 1.6.11. For an example in $\mathbf{R}^4$ (which we can't draw), consider[5]

$$\mathbf{v}'' = \begin{bmatrix} 1 \\ 2 \\ 1 \\ -2 \end{bmatrix}, \quad \mathbf{w}'' = \begin{bmatrix} 3 \\ -2 \\ 1 \\ 1 \end{bmatrix}.$$

The distance between them is the length of the difference

$$\mathbf{v}'' - \mathbf{w}'' = \begin{bmatrix} -2 \\ 4 \\ 0 \\ -3 \end{bmatrix},$$

which is $\sqrt{(-2)^2 + 4^2 + 0^2 + (-3)^2} = \sqrt{4 + 16 + 9} = \sqrt{29} \approx 5.3852.$ ■

Remark 1.6.12. It is a fact of experience that "the shortest distance between two points is a straight line". This inspires an insight about vectors. To explain it, consider points P, Q, R in $\mathbf{R}^3$ as in Figure 1.6.6.

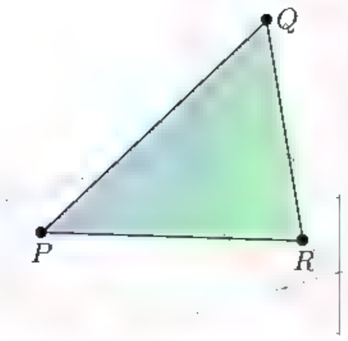


FIGURE 1.6.6. Traveling from P to R via Q

If we travel from point P to point R by first going from P straight to Q and then from Q straight to R, we conclude that (distance from P to Q) + (distance from Q to R) $\geq$ (distance from P to R). In the language of displacement vectors from Example 1.4.1, if the vectors corresponding to the points P, Q, R are respectively denoted as $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ then this relation among distances says

$$\|\mathbf{v}_2 - \mathbf{v}_1\| + \|\mathbf{v}_3 - \mathbf{v}_2\| \geq \|\mathbf{v}_3 - \mathbf{v}_1\|.$$

But this final inequality is meaningful to consider for vectors in $\mathbf{R}^n$ for *any* n. In other words:

Question: for any n, do all n-vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ satisfy the inequality

$$\|\mathbf{v}_3 - \mathbf{v}_1\| \leq \|\mathbf{v}_2 - \mathbf{v}_1\| + \|\mathbf{v}_3 - \mathbf{v}_2\|? \tag{1.6.1}$$

This Question is *inspired* by experience in the familiar cases $n = 2, 3$ and so is thereby geometrically "plausible" to contemplate for any n. Due to the motivation from going two ways around a triangle, the inequality (1.6.1) is called the *triangle inequality*. But is it true for all n?

In the special case that $\mathbf{v}_2 = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$ is the n-vector whose entries are all equal to 0, if we define $\mathbf{x} = -\mathbf{v}_1$ and $\mathbf{y} = \mathbf{v}_3$ then it is the same to ask if

$$\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\| \tag{1.6.2}$$

---

[5] Notation such as $\mathbf{v}''$ and $\mathbf{w}''$ here does *not* denote derivatives (it would make no sense). In math, such dashes denote additional versions of some common structure (such as vectors here, or t, t', t'' for time variables, etc.) when not functions.

for all n-vectors $\mathbf{x}, \mathbf{y}$. This is an *equivalent* formulation of (1.6.1), as we see by substituting $\mathbf{v}_2 - \mathbf{v}_1$ for $\mathbf{x}$ and $\mathbf{v}_3 - \mathbf{v}_2$ for $\mathbf{y}$ in (1.6.2) (since then $\mathbf{x}+\mathbf{y} = \mathbf{v}_2 - \mathbf{v}_1 + \mathbf{v}_3 - \mathbf{v}_2 = \mathbf{v}_3 - \mathbf{v}_1$). For those who are interested, Theorem 2.5.1 affirmatively answers the Question about (1.6.1) by using the formulation (1.6.2).

Example 1.6.13. An important application where averaging of n-vectors and distance of n-vectors arise in essential ways is the *K-means algorithm*, an "unsupervised learning" algorithm that is used by computers to divide (or "cluster") $\mathbf{R}^n$-data into K batches. It is explained by Andrew Ng in this video using just the concepts we have discussed so far (he writes "$\mathbf{x} \in \mathbf{R}^n$" as shorthand for "$\mathbf{x}$ is an n-vector"); check it out! This algorithm is widely used (often with $n > 3$), such as in computer vision [**CBJD**], market segmentation [**KW**, Ch. 5], and even fantasy sports, though other clustering algorithms are preferred for some applications; e.g., to satisfy driver preferences, Uber uses "spectral biclustering" [**FF**] based on the Singular Value Decomposition introduced in Section 27.3. ■

1.7. Unit vectors.

Definition 1.7.1. The *zero vector* in $\mathbf{R}^n$ is $\mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, and a *unit vector* is a vector with length 1.

Always $\|\mathbf{v}\| \geq 0$, and $\|\mathbf{v}\| = 0$ precisely when $\mathbf{v} = \mathbf{0}$. If $\mathbf{v} \neq \mathbf{0}$ then dividing $\mathbf{v}$ by its length (i.e., multiplying by the scalar $\dfrac{1}{\|\mathbf{v}\|} > 0$) yields a unit vector "pointing in the same direction" as $\mathbf{v}$.

It should feel reasonable that to stretch or shrink a nonzero n-vector $\mathbf{v}$ to become a unit vector, we should divide it by its length (really: multiply $\mathbf{v}$ by the scalar $1/\|\mathbf{v}\|$). Let's visualize this in $\mathbf{R}^3$, then work out an example in $\mathbf{R}^3$, and finally use general algebraic definitions to understand it for $\mathbf{R}^n$ with *any* n.

If $\mathbf{v}$ is a nonzero vector in $\mathbf{R}^3$ and c is any *positive* real number, then the two vectors $c\mathbf{v}$ and $\mathbf{v}$ point in the same direction. For example $7\mathbf{v}$ points in the same direction but it is 7 times as long (akin to changing the units of distance by a factor of 7), whereas $(2/3)\mathbf{v}$ points in the same direction but is shorter by a factor of $2/3$. Let's see in an example what happens when we divide by the length:

Example 1.7.2. Let $\mathbf{v} = \begin{bmatrix} -2 \\ 1 \\ 2 \end{bmatrix}$, so $\|\mathbf{v}\| = \sqrt{4+1+4} = \sqrt{9} = 3$. Hence, dividing by the length yields $\dfrac{1}{3}\mathbf{v} = \begin{bmatrix} -2/3 \\ 1/3 \\ 2/3 \end{bmatrix}$, which has length $\sqrt{4/9+1/9+4/9} = \sqrt{9/9} = 1$ as desired. ■

In general the length of $c\mathbf{v}$ for $c > 0$ is $c\|\mathbf{v}\|$, so in order that $c\mathbf{v}$ be a unit vector the condition is precisely that $c\|\mathbf{v}\| = 1$, which is to say $c = 1/\|\mathbf{v}\|$. In other words, the scalar multiple

$$\frac{1}{\|\mathbf{v}\|}\mathbf{v}$$

of $\mathbf{v}$ is indeed the unique unit vector pointing in the *same* direction as $\mathbf{v}$. (In the *opposite* direction we have the unit vector $(-1/\|\mathbf{v}\|)\,\mathbf{v}$.)

In a situation where only the *direction* of a vector matters, it is often useful to describe that direction by using a unit vector. In Section 11.3 we will discuss the technique of *gradient descent* that is a ubiquitous tool for solving multivariable optimization problems in many applied settings, such as for backpropagation in machine learning (which we discuss in Appendix G) and linear regression (which we introduce in Chapter 7). This technique uses the unit vector in the direction of a "gradient vector" (a type of multivariable derivative) of a function: see Theorem 11.3.2.

Example 1.7.3. Consider a map where the positive y-axis points north and the positive x-axis points east. The unit vector that points due west is $\begin{bmatrix} -1 \\ 0 \end{bmatrix}$. What is the unit vector that is pointing in the exactly southwest direction? To figure this out, let's first give *some* vector pointing in exactly the southwest direction, and then divide by its length to get a unit vector.

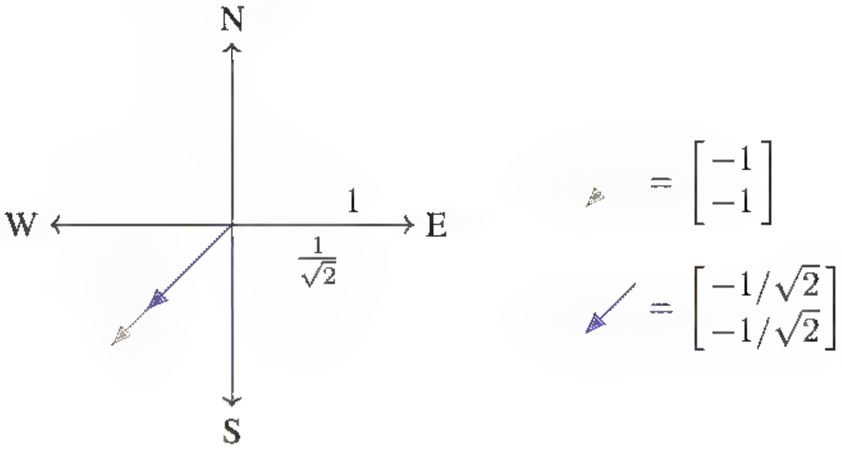

N
W
E
S
1
$\frac{1}{\sqrt{2}}$
$= \begin{bmatrix} -1 \\ -1 \end{bmatrix}$
$= \begin{bmatrix} -1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}$

FIGURE 1.7.1. A unit vector pointing southwest

A vector in the southwest direction is $\begin{bmatrix} -1 \\ -1 \end{bmatrix}$, as in Figure 1.7.1. This isn't a unit vector: its length is $\sqrt{(-1)^2 + (-1)^2} = \sqrt{2}$. Dividing by its length gives a unit vector in the same direction:

$$\begin{bmatrix} -1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}.$$

■

Remark 1.7.4. Two themes of this book, the first of which has been illustrated in this chapter, are:

(i) geometric concepts in $\mathbf{R}^3$ can be usefully generalized to $\mathbf{R}^n$ for any n (e.g., length and distance; we'll discuss angles in Chapter 2),

(ii) such generalizations provide visual insight into many real-world situations, such as: physical systems with many parts (cars, bridges, circuits, etc.), friendship networks, economic forecasts, Netflix recommendations, neural networks, and so on (as we'll discuss later, such as in Example 4.1.9, Example 14.2.3 Section 16.4, Section 21.1, Example 21.6.3, and Appendix G).

Geometry in $\mathbf{R}^n$ for big n (to be discussed and used more fully later on) is *an entirely human creation and doesn't require any universal interpretation in terms of the world around us.*

There's a lot of nonsense written about a mystical "fourth dimension", or saying it is time, etc. It is all irrelevant baloney which you should ignore. There are *zillions* of real-world problems involving many unknowns; in all of them it is very convenient to use $\mathbf{R}^n$ for various (often big) values of n. There is no need to assign cosmic meaning to a "dimension" irrespective of a problem at hand.

The geometric language of vectors and distance (and other notions to come later) in $\mathbf{R}^n$ is a powerful psychological device, with which human beings can harness their visualization skills to "feel" what is going on and think efficiently when working with n-vectors for any value of n whatsoever.

Chapter 1 highlights (link to highlights in next chapter)

Notation	**Meaning**	**Location in text**
$\mathbf{R}^n$	the collection of all n-vectors	Definition 1.1.2
$\mathbf{v} + \mathbf{w}$	the sum of n-vectors $\mathbf{v}$ and $\mathbf{w}$	Definition 1.3.1
$c\,\mathbf{v}$	multiple of scalar c against n-vector $\mathbf{v}$	Definition 1.3.2
$\lVert\mathbf{v}\rVert$	length (or magnitude) of an n-vector $\mathbf{v}$	Definition 1.6.1

Concept	**Meaning**	**Location in text**
n-vector	list of n real numbers, written vertically as $\begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$	Definition 1.1.2
scalar	a real number (the type of number arising in calculus)	Definition 1.1.4
vector addition	operation on n-vectors adding corresponding entries (**cannot** add n-vector to m-vector if $n \neq m$)	Definition 1.3.1
scalar multiplication	operation on a scalar c and n-vector multiplying all entries by c	Definition 1.3.2
linear combination	for given n-vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$, it is any n-vector of the form $c_1\mathbf{v}_1 + \cdots + c_k\mathbf{v}_k$ for scalars $c_1, \dots, c_k$	Definition 1.3.4
convex combination	for given n-vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$, it is $c_1\mathbf{v}_1 + \cdots + c_k\mathbf{v}_k$ with $0 \leq c_1, \dots, c_k \leq 1$ and $\sum_{j=1}^{k} c_j = 1$	Example 1.3.8
length	for $\mathbf{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$ it is the scalar $\sqrt{v_1^2 + \cdots + v_n^2}$	Definition 1.6.1
distance	for n-vectors $\mathbf{x}$ and $\mathbf{y}$, it is $\lVert\mathbf{x} - \mathbf{y}\rVert = \lVert\mathbf{y} - \mathbf{x}\rVert$	Definition 1.6.4 (Figure 1.6.1 for $n = 3$)
unit vector	an n-vector whose length equals 1	Definition 1.7.1

Result	**Meaning**	**Location in text**
parallelogram law	visualization of vector addition in $\mathbf{R}^2$ and $\mathbf{R}^3$ (has no analogue in $\mathbf{R}^n$ for $n > 3$, but is useful mental guide for what to expect with vector addition for general n)	Example 1.3.3
displacement	visualizes going from tip of n-vector P to tip of n-vector Q via adding to P the vector difference $Q - P$	Example 1.4.1
properties of vector addition and scalar multiplication	both operations satisfy "all" meaningful analogues of properties of addition and multiplication of ordinary numbers	Theorem 1.5.2

Skill	**Location in text**
add n-vectors and multiply them by scalars	Example 1.3.5
know how linear combinations of n-vectors arise (e.g., with grades) and how to visualize convex combinations when $n = 2$	Section 1.4
compute length of and distance between n-vectors	Examples 1.6.2, 1.6.10, 1.6.11
compute unit vector in same direction as a given nonzero n-vector	Examples 1.7.2 and 1.7.3

2. Vector geometry in $\mathbf{R}^n$ and correlation coefficients

Near the end of Chapter 1 we defined "length" for vectors in $\mathbf{R}^n$. Geometric reasoning with vectors can go much further, and to develop "geometric intuition" in $\mathbf{R}^n$ it is best to begin in the more familiar setting of $\mathbf{R}^3$. In this chapter we first show how many questions in 3-dimensional geometry that can be answered via trigonometry are a lot easier to answer with the language of vectors. That motivates how we can use geometric language to work with vectors in new ways. In particular, we generalize some considerations in 3-dimensional space to $\mathbf{R}^n$ with $n > 3$; this will be very fruitful in our later study and use of vectors.

By the end of this chapter, you should be able to do the following for vectors in any $\mathbf{R}^n$:

- compute the dot product between vectors;
- compute angles between vectors, and check for perpendicularity, by using the dot product;
- compute the correlation coefficient using n-vectors, and recognize graphically from a plot of n data points the significance of positive and negative correlation coefficients.

2.1. Angles. In Section 1.6 we defined the notion of *length* of a vector. In $\mathbf{R}^2$ this expressed the Pythagorean Theorem, in $\mathbf{R}^3$ it was deduced from instances of the planar case (see Example 1.6.5), and for n-vectors we used the analogous formula as a *definition*. We now carry out a similar discussion for angles, building on the Law of Cosines in place of the Pythagorean Theorem:

Proposition 2.1.1. The angle $0° \leq \theta \leq 180°$ between nonzero 2-vectors $\mathbf{a} = (a_1, a_2)$ and $\mathbf{b} = (b_1, b_2)$ satisfies

$$\cos\theta = \frac{a_1b_1 + a_2b_2}{\|\mathbf{a}\|\,\|\mathbf{b}\|}. \tag{2.1.1}$$

In other words, the angle θ is equal to $\arccos$ (sometimes denoted $\cos^{-1}$) applied to the right side of (2.1.1).

This is the Law of Cosines in disguise. Indeed, consider the triangle with vertices at the origin $(0, 0)$, (a_1, a_2), and (b_1, b_2), as in Figure 2.1.1.

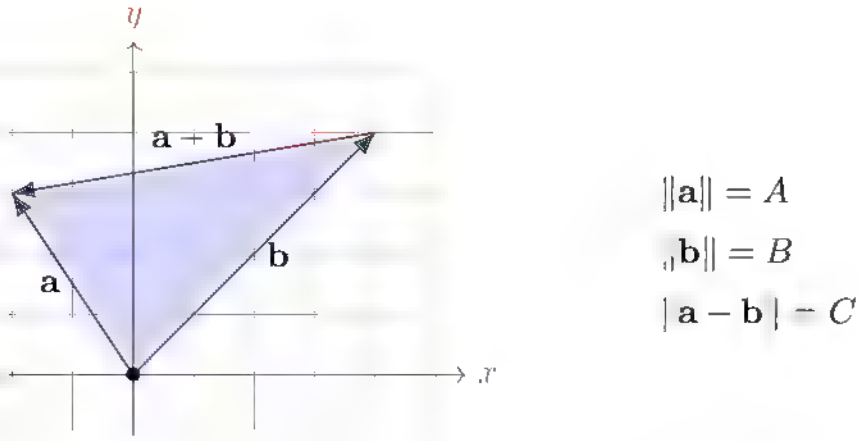


FIGURE 2.1.1. Triangle in $\mathbf{R}^2$ with vertices at $\mathbf{0}$, $\mathbf{a}$, and $\mathbf{b}$

Let $A = \|\mathbf{a}\|$ and $B = \|\mathbf{b}\|$ denote the lengths of the sides joining $(0, 0)$ to (a_1, a_2) and (b_1, b_2) respectively. The third side has length C given by that of the difference vector $\mathbf{a} - \mathbf{b}$ as shown in Figure 2.1.1, so $C = \|\mathbf{a} - \mathbf{b}\|$. The Law of Cosines says $C^2 = A^2 + B^2 - 2AB\cos\theta$, so

$$\cos\theta = \frac{A^2 + B^2 - C^2}{2AB}. \tag{2.1.2}$$

This numerator $A^2 + B^2 - C^2 = (a_1^2 + a_2^2) + (b_1^2 + b_2^2) - ((a_1 - b_1)^2 + (a_2 - b_2)^2)$ expands out to

$$a_1^2 + a_2^2 + b_1^2 + b_2^2 - ((a_1^2 - 2a_1b_1 + b_1^2) + (a_2^2 - 2a_2b_2 + b_2^2)) = 2(a_1b_1 + a_2b_2).$$

Plugging that into the numerator of (2.1.2) and cancelling the common factor of 2 in the top and bottom yields (2.1.1).

Example 2.1.2. Let's find a nonzero vector in $\mathbf{R}^2$ *perpendicular* to $\mathbf{u} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$; i.e., a nonzero 2-vector $\mathbf{v}$ that makes an angle of $\pi/2$ radians (or 90°) with $\begin{bmatrix} 3 \\ 4 \end{bmatrix}$, such as shown in Figure 2.1.2.

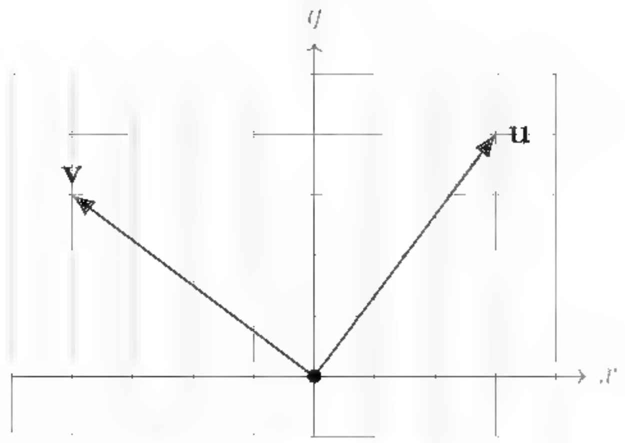


FIGURE 2.1.2. A vector $\mathbf{v}$ perpendicular to a given vector $\mathbf{u}$

Since $\cos(\pi/2) = 0$, by (2.1.1) we seek a nonzero 2-vector $\mathbf{v} = \begin{bmatrix} a \\ b \end{bmatrix}$ so that $\dfrac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} = 0$, which is tantamount to the numerator being 0. In other words, we seek a, b not both zero with

$$0 = \mathbf{u} \cdot \mathbf{v} = 3a + 4b.$$

So we just pick whatever value $b \neq 0$ we like, and then define a to be $-4b/3$. For example, by setting $b = 3$ we get $a = -4$, so the vector $\begin{bmatrix} -4 \\ 3 \end{bmatrix}$ is perpendicular to $\begin{bmatrix} 3 \\ 4 \end{bmatrix}$ (as are $\begin{bmatrix} -8 \\ 6 \end{bmatrix}, \begin{bmatrix} 4 \\ -3 \end{bmatrix}, \begin{bmatrix} -4/3 \\ 1 \end{bmatrix}$, and more generally $\begin{bmatrix} -4y/3 \\ y \end{bmatrix}$ for any $y \neq 0$). ■

Theorem 2.1.3. The angle $0° \leq \theta \leq 180°$ between two nonzero 3-vectors $\mathbf{a} = (a_1, a_2, a_3)$ and $\mathbf{b} = (b_1, b_2, b_3)$ satisfies

$$\cos\theta = \frac{a_1 b_1 + a_2 b_2 + a_3 b_3}{\|\mathbf{a}\|\|\mathbf{b}\|}. \tag{2.1.3}$$

In other words, the angle θ is equal to arccos (sometimes denoted $\cos^{-1}$) applied to the right side of (2.1.3).

Remark 2.1.4. In Section 2.5, for those who are interested, we justify (2.1.3) using the Law of Cosines.

We have not seen the numerator $a_1 b_1 + a_2 b_2 + a_3 b_3$ before, and we give it a special name: the *dot product*, denoted $\mathbf{a} \cdot \mathbf{b}$. Thus $\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + a_3 b_3$ is a scalar, not a vector.

Example 2.1.5. Let's determine the angle between the vectors $\mathbf{u} = \begin{bmatrix} 3 \\ 0 \\ 4 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} 1 \\ 3 \\ -2 \end{bmatrix}$ in Figure 2.1.3.

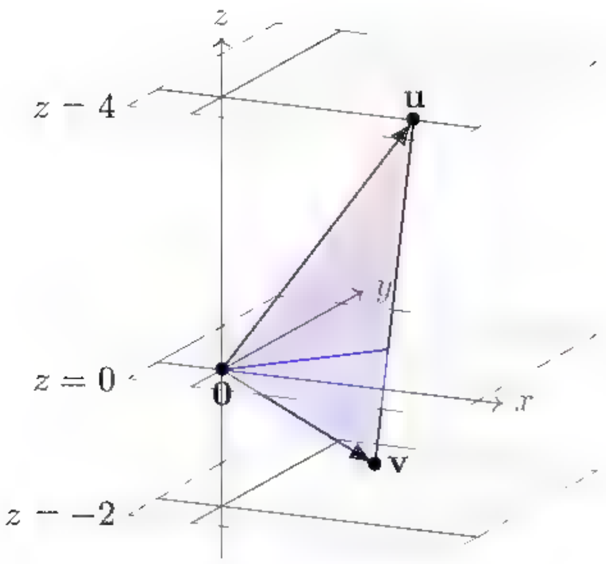


FIGURE 2.1.3. Angle between $\mathbf{u}$ and $\mathbf{v}$ is that of the purple triangle at the corner $\mathbf{0}$

We compute the dot product and lengths

$$\mathbf{u}\cdot\mathbf{v} = 3+0-8 = -5, \quad \|\mathbf{u}\| = \sqrt{3^2+0^2+4^2} = 5, \quad \|\mathbf{v}\| = \sqrt{1^2+3^2+(-2)^2} = \sqrt{14}.$$

Therefore, the angle θ satisfies $\cos(\theta) = -5/(5\sqrt{14}) = -1/\sqrt{14}$, so $\theta = \arccos(-1/\sqrt{14}) \approx 105.5^\circ$. ■

The preceding in $\mathbf{R}^2$ and $\mathbf{R}^3$ motivates how to *define* appropriate concepts with n-vectors for any n:

Definition 2.1.6. Consider n-vectors $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$.

(i) The *dot product* of $\mathbf{x}$ and $\mathbf{y}$ is defined to be the scalar

$$\mathbf{x}\cdot\mathbf{y} = x_1y_1 + x_2y_2 + \cdots + x_ny_n = \sum_{i=1}^{n} x_iy_i.$$

(The dot product is only defined if the two vectors are n-vectors for the same value of n.)

(ii) The *angle* θ between two nonzero n-vectors $\mathbf{x}, \mathbf{y}$ is defined by the formula

$$\cos(\theta) = \frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\,\|\mathbf{y}\|} \tag{2.1.4}$$

with $0^\circ \leq \theta \leq 180^\circ$. (For emphasis: $\mathbf{x}$ and $\mathbf{y}$ must be nonzero n-vectors for a common n.)

(iii) When $\mathbf{x}\cdot\mathbf{y} = 0$ (same as $\theta = 90^\circ$ if $\mathbf{x}, \mathbf{y} \neq \mathbf{0}$), we say $\mathbf{x}$ and $\mathbf{y}$ are *perpendicular*; the word *orthogonal* is often used for this ("orthogōnios" is ancient Greek for "right-angled"), though only rarely[6] at the U.S. Supreme Court.

Always remember that the dot product of vectors is a scalar (it is *not* a vector).

Example 2.1.7. Dot products are easy to compute. For an example with $\mathbf{R}^3$,

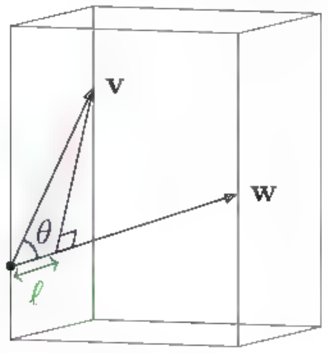
$$\begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ -2 \\ 4 \end{bmatrix} = 2\cdot 3 + 1\cdot(-2) + 3\cdot 4 \;=\; 6 + -2 + 12 = 16.$$

---

[6] An audio version begins at 22:29 in the Oral Argument link here. The meaning of orthogonality there is also the reason for the name of the field bioorthogonal chemistry, whose development by Carolyn Bertozzi at Stanford earned her a 2022 Nobel Prize in Chemistry.

Similarly with $\mathbf{R}^4$, $\begin{bmatrix} 3 \\ 1 \\ 0 \\ 1/2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -2 \\ 3 \\ 6 \end{bmatrix} = 3 - 2 + 0 + 3 = 4.$ ■

Remark 2.1.8. The visual meaning of $\mathbf{v} \cdot \mathbf{w}$ is addressed in Chapter 6, but let's state it now for nonzero 3-vectors $\mathbf{v}$ and $\mathbf{w}$ making an *acute* angle θ as in Figure 2.1.4. For the (colored) right triangle with hypotenuse $\mathbf{v}$ and a leg along the line through $\mathbf{w}$, the length ℓ of that leg is $\|\mathbf{v}\| \cos\theta = (\|\mathbf{v}\|\|\mathbf{w}\| \cos\theta)/\|\mathbf{w}\| = (\mathbf{v} \cdot \mathbf{w})/\|\mathbf{w}\|$, so $\mathbf{v} \cdot \mathbf{w} = \|\mathbf{w}\|\ell$. This equation is a special case of Proposition 6.1.1.



$$\ell = \|\mathbf{v}\| \cos\theta = \frac{\|\mathbf{v}\| \, \|\mathbf{w}\| \cos\theta}{\|\mathbf{w}\|} = \frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{w}\|}$$

FIGURE 2.1.4. Right triangle formed by two 3-vectors making an acute angle

Inspired by Figure 2.1.4, you might wonder: why not geometrically *define* $\mathbf{v} \cdot \mathbf{w}$ to be $\|\mathbf{w}\|$ times the length of that leg? This would be terrible, for a few reasons: (i) it doesn't (yet) make sense for $n > 3$ (but the dot product with $n > 3$ will later pervade a vast array of real-world applications, due to geometric results in Chapter 6), (ii) it treats $\mathbf{v}$ and $\mathbf{w}$ in very different roles whereas in fact $\mathbf{v} \cdot \mathbf{w} = \mathbf{w} \cdot \mathbf{v}$ (see Theorem 2.2.1), (iii) it makes important algebraic properties of the dot product (in Theorem 2.2.1) hard to understand.

Remark 2.1.9. We emphasize again that it *does not make sense* to compute a dot product $\mathbf{v} \cdot \mathbf{w}$ if $\mathbf{v}$ is an n-vector and $\mathbf{w}$ is an m-vector with $n \neq m$. If you ever find yourself trying to compute the dot product of a vector in $\mathbf{R}^4$ and a vector in $\mathbf{R}^3$, for example, you have made a mistake earlier in your work.

Example 2.1.10. Here is an example of dot products that arises in the study of consumer satisfaction in microeconomics. Suppose there are n goods available to consumers in some small part of the economy. For example, if the goods under consideration are rice and beans then $n = 2$. Let p_i denote the price of the ith good (so $p_i \geq 0$) and let x_i denote the quantity of the ith good that a person chooses to purchase (so $x_i \geq 0$). These assemble into a "price n-vector" $\mathbf{p}$ and a "quantity n-vector" $\mathbf{x}$ respectively. In the case of rice and beans, we may have

$$\mathbf{p} = (\text{price of rice in dollars per pound}, \text{price of beans in dollars per pound}),$$
$$\mathbf{x} = (\text{pounds of rice purchased}, \text{pounds of beans purchased}).$$

Note that the dot product

$$\mathbf{p} \cdot \mathbf{x} = \sum p_i x_i$$

is the total amount spent by the consumer.

In economics, consumer satisfaction for a given vector $\mathbf{x}$ of purchased goods is measured by a number $u(\mathbf{x})$, where $u : \mathbf{R}^n \to \mathbf{R}$ is a *utility function* (whose precise definition depends on the specific economic circumstances). If a consumer is constrained not to spend more than a given amount $w \geq 0$ of wealth, which is to say $\mathbf{p} \cdot \mathbf{x} \leq w$, then maximizing customer satisfaction subject to such a budgetary constraint is expressed mathematically by

$$f(\mathbf{p}, w) = \text{the } n\text{-vector } \mathbf{x} \text{ maximizing } u(\mathbf{x}) \text{ subject to the condition } \mathbf{p} \cdot \mathbf{x} \leq w.$$

Under suitable assumptions on u (studied in consumer theory) there is exactly one such maximizer $\mathbf{x}$, so the n-vector $f(\mathbf{p}, w)$ makes sense unambiguously and is called the *optimal quantity vector*. For a given $\mathbf{p}$, a consumer with wealth at most w should purchase the goods in accordance with the optimal quantity vector $f(\mathbf{p}, w)$ to maximize their satisfaction. The function f is called the *Marshallian demand function* in economics (and by design, it depends on the specific utility function u). ■

The notion of angle in Definition 2.1.6(ii) above is a *definition* in $\mathbf{R}^n$ for general n: it is motivated by (2.1.3) in the case $n = 3$ (as that ensures Definition 2.1.6(ii) recovers the familiar notion of angle when $n = 3$), but for general n there is nothing to "physically justify" in Definition 2.1.6(ii). The real content in making this definition for general n is that (as you will learn with experience) this notion of angle behaves like our visual experience in $\mathbf{R}^2$ and $\mathbf{R}^3$ (see the beginning of Example 2.2.4 for an illustration) and so provides useful visual guidance with n-vectors for any n.

Remark 2.1.11. If we multiply either $\mathbf{x}$ or $\mathbf{y}$ by -1 in the definition of angle then the dot product is replaced with its negative (e.g., $\mathbf{x} \cdot (-\mathbf{y}) = -(\mathbf{x} \cdot \mathbf{y})$) but the lengths don't change, so the cosine of the angle is replaced with its negative. That corresponds to passing to the *supplementary angle* $180^\circ - \theta$, as indicated in Figure 2.1.5 below, since $\cos(180^\circ - \theta) = -\cos(\theta)$.

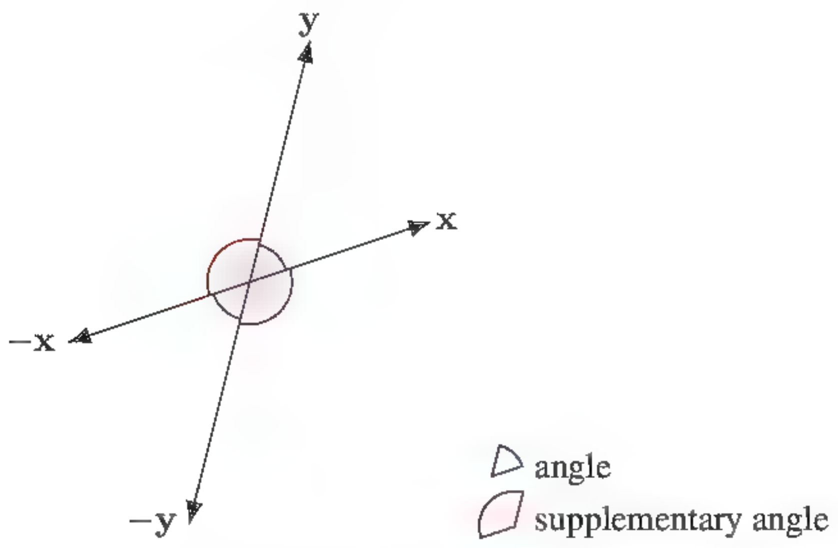


FIGURE 2.1.5. Angle and supplementary angle

Whenever we speak of an angle *between two lines through the origin*, there is always an ambiguity (when they're not perpendicular) of whether we want the acute angle between them or the (supplementary) obtuse angle between them. This corresponds to the fact that when we set it up as a vector problem, we have to choose a *direction* along each line (coming out of the intersection point). Depending on the choice, we will get the acute or the obtuse angle. Flipping the choice of direction corresponds to multiplying the choice of vector along the line by -1, and so is consistent with our observation above with the formula (2.1.4) that multiplying $\mathbf{x}$ or $\mathbf{y}$ by -1 causes the angle to switch to its supplement.

Example 2.1.12. In $\mathbf{R}^2$, any pair of vectors of the form $\begin{bmatrix} a \\ b \end{bmatrix}$ and $\begin{bmatrix} -b \\ a \end{bmatrix}$ are perpendicular since $\begin{bmatrix} a \\ b \end{bmatrix} \cdot \begin{bmatrix} -b \\ a \end{bmatrix} = -ab + ba = 0$. We saw an instance of this in Example 2.1.2 with $a = 3$ and $b = 4$. ■

Example 2.1.13. Here is an interesting geometric application of the ability to compute angles between vectors in $\mathbf{R}^3$ via dot products: what is the acute angle between two diagonal lines of a cube? Consider the "unit cube" in $\mathbf{R}^3$ consisting of points (x, y, z) with $0 \leq x, y, z \leq 1$, as in Figure 2.1.6. The diagonals are also diagonals of a rectangle with side lengths 1 and $\sqrt{2}$ as shown, so the problem can be analyzed via plane geometry. We will solve it using dot products of displacement vectors.

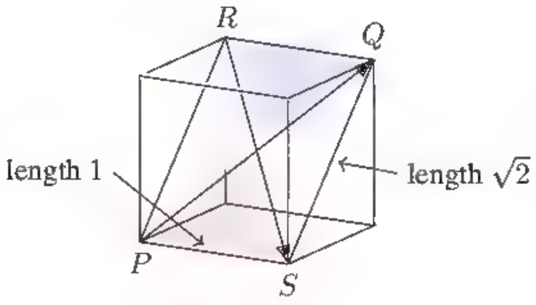


FIGURE 2.1.6. Unit cube in $\mathbf{R}^3$

The 8 vertices are

$$P=(0,0,0), S=(1,0,0), (0,1,0), (0,0,1), (1,1,0), (1,0,1), R=(0,1,1), Q=(1,1,1).$$

Two of the diagonals are the segments PQ and RS, with corresponding displacement vectors (up to sign, keeping in mind Remark 2.1.11)

$$\mathbf{x}=\begin{bmatrix}1\\1\\1\end{bmatrix}-\begin{bmatrix}0\\0\\0\end{bmatrix}=\begin{bmatrix}1\\1\\1\end{bmatrix},\ \mathbf{y}=\begin{bmatrix}1\\0\\0\end{bmatrix}-\begin{bmatrix}0\\1\\1\end{bmatrix}=\begin{bmatrix}1\\-1\\-1\end{bmatrix}.$$

The angle θ between these satisfies $\cos(\theta)=\dfrac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}=\dfrac{-1}{\sqrt{3}\sqrt{3}}=-\dfrac{1}{3}$. This is obtuse (as its cosine is negative), so via Remark 2.1.11 we flip the sign to get that the acute angle is $\arccos(1/3)\approx 70.5^{\circ}$. ■

Remark 2.1.14 (optional). Now that we have some experience with Definition 2.1.6(ii), we should acknowledge a subtlety: does it even *make sense* to say the right side of (2.1.4) is the cosine of an angle? The values of cosine lie between -1 and 1, so if that right side could ever equal $8/7$ (or anything else outside the interval $[-1,1]$) then it could not be written as the cosine of anything!

The graph of cosine across the closed interval of angles from 0° to 180° (or equivalently, the closed interval of radians from 0 to π) attains exactly the values between $\cos 0^{\circ}=1$ and $\cos 180^{\circ}=-1$, due to the graph of $\cos x$ as in Figure 2.1.7.

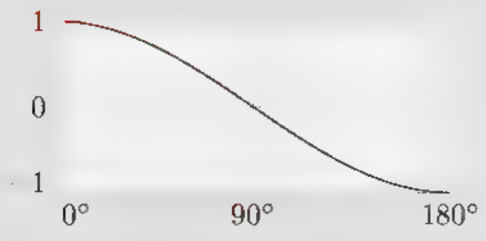


FIGURE 2.1.7. The graph of $\cos x$ for angles x from 0° to 180°

Note that $\cos x$ attains each value between -1 and 1 *without* repetition on $[0^{\circ},180^{\circ}]$: different angles in this interval have different cosine values, as we see since the graph is strictly decreasing across the interval. Thus, for Definition 2.1.6(ii) to make sense (necessarily unambiguously) as a definition it must be shown that the right side of (2.1.4) always lies between -1 and 1. That this holds is not at all obvious; it is part of Theorem 2.3.2, which rests on properties of dot products.

2.2. Properties of dot products. The following properties of the dot product can be established by direct algebraic manipulation with the formula that defines the dot product:

Theorem 2.2.1. For any n-vectors $\mathbf{v}$, $\mathbf{w}$, $\mathbf{w}_1$, and $\mathbf{w}_2$, the following hold:

(i) $\mathbf{v}\cdot\mathbf{w} = \mathbf{w}\cdot\mathbf{v}$,

(ii) $\mathbf{v}\cdot\mathbf{v} = \|\mathbf{v}\|^2$,

(iii) $\mathbf{v}\cdot(c\mathbf{w}) = c(\mathbf{v}\cdot\mathbf{w})$ for any scalar c, and $\mathbf{v}\cdot(\mathbf{w}_1+\mathbf{w}_2) = \mathbf{v}\cdot\mathbf{w}_1 + \mathbf{v}\cdot\mathbf{w}_2$.

(iii$'$) Combining both rules in (iii), for any scalars c_1, c_2 we have

$$\mathbf{v}\cdot(c_1\mathbf{w}_1 + c_2\mathbf{w}_2) = c_1(\mathbf{v}\cdot\mathbf{w}_1) + c_2(\mathbf{v}\cdot\mathbf{w}_2).$$

To illustrate the type of work which explains the general properties in Theorem 2.2.1, assertion (i) amounts to the calculation $\mathbf{v}\cdot\mathbf{w} = \sum_{i=1}^n v_i w_i = \sum_{i=1}^n w_i v_i = \mathbf{w}\cdot\mathbf{v}$. (For those who are interested, in Remark 2.5.2 the algebraic Theorem 2.2.1 and the geometric Theorem 2.1.3 – in effect, the Law of Cosines – are used together to give a nifty proof of the Law of Sines.)

Example 2.2.2. Consider the vectors $\mathbf{v} = \begin{bmatrix}1\\3\\2\end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix}-2\\4\\3\end{bmatrix}$ in $\mathbf{R}^3$. Illustrating (i) above, we have

$$\mathbf{v}\cdot\mathbf{w} = \begin{bmatrix}1\\3\\2\end{bmatrix}\cdot\begin{bmatrix}-2\\4\\3\end{bmatrix} = 1\cdot(-2)+3\cdot4+2\cdot3 = 16,\ \mathbf{w}\cdot\mathbf{v} = \begin{bmatrix}-2\\4\\3\end{bmatrix}\cdot\begin{bmatrix}1\\3\\2\end{bmatrix} = (-2)\cdot1+4\cdot3+3\cdot2 = 16,$$

so $\mathbf{v}\cdot\mathbf{w} = \mathbf{w}\cdot\mathbf{v}$. Illustrating (ii) above, we have $\mathbf{v}\cdot\mathbf{v} = 1^2+3^2+2^2 = 14$, $\|\mathbf{v}\| = \sqrt{1^2+3^2+2^2} = \sqrt{14}$, so $\|\mathbf{v}\|^2 = 14 = \mathbf{v}\cdot\mathbf{v}$. Finally, to illustrate (iii$'$) above, we have

$$\begin{bmatrix}1\\3\\2\end{bmatrix}\cdot\left(5\begin{bmatrix}-2\\4\\3\end{bmatrix}+2\begin{bmatrix}3\\1\\2\end{bmatrix}\right) = \begin{bmatrix}1\\3\\2\end{bmatrix}\cdot\left(\begin{bmatrix}-10\\20\\15\end{bmatrix}+\begin{bmatrix}6\\2\\4\end{bmatrix}\right) = \begin{bmatrix}1\\3\\2\end{bmatrix}\cdot\begin{bmatrix}-4\\22\\19\end{bmatrix} = -4+66+38 = 100$$

and

$$5\begin{bmatrix}1\\3\\2\end{bmatrix}\cdot\begin{bmatrix}-2\\4\\3\end{bmatrix} + 2\begin{bmatrix}1\\3\\2\end{bmatrix}\cdot\begin{bmatrix}3\\1\\2\end{bmatrix} = 5\cdot16+2\cdot10 = 80+20 = 100.$$

■

Example 2.2.3. Let's see that with the definition of angle we have given, for any nonzero n-vector $\mathbf{x}$ and scalar $c \neq 0$, the angle between $\mathbf{x}$ and $c\mathbf{x}$ behaves as we would expect from experience in $\mathbf{R}^2$ and $\mathbf{R}^3$:

- 0 radians (or 0°) if $c > 0$ (in this case $\mathbf{x}$ and $c\mathbf{x}$ "point in the same direction");
- π radians (or 180°) if $c < 0$ (in this case $\mathbf{x}$ and $c\mathbf{x}$ "point in opposite directions").

For instance, in Figure 2.2.1 we use scalars $c_1 > 0$ and $c_2 < 0$:

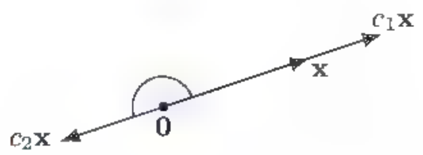


FIGURE 2.2.1. Vectors making angles 0 and π with a nonzero vector $\mathbf{x}$

The reason this works is just plugging into the definition, using Theorem 2.2.1(ii),(iii), and recalling that $c\mathbf{x}$ has length $|c|\|\mathbf{x}\|$. We get $\cos(\theta) = \frac{(c\mathbf{x})\cdot\mathbf{x}}{\|c\mathbf{x}\|\|\mathbf{x}\|} = \frac{c(\mathbf{x}\cdot\mathbf{x})}{|c|\|\mathbf{x}\|\|\mathbf{x}\|} = \frac{c(\mathbf{x}\cdot\mathbf{x})}{|c|\|\mathbf{x}\|^2}$ yet $\|\mathbf{x}\|^2 = \mathbf{x}\cdot\mathbf{x}$, so cancellation leaves us with $c/|c|$; this is 1 when $c > 0$ and -1 when $c < 0$. Hence, if $c > 0$ then the angle is $\arccos(1) = 0^\circ$ and if $c < 0$ then it is $\arccos(-1) = 180^\circ$ as desired.

In fact, it turns out that this is the *only* situation in which such angles can arise. That is, if $\mathbf{x}, \mathbf{y}$ are nonzero vectors in $\mathbf{R}^n$ for which the angle between them is either 0° or 180° (equivalently, $\mathbf{x} \cdot \mathbf{y} = \pm\|\mathbf{x}\|\|\mathbf{y}\|$) then we claim that *necessarily* $\mathbf{y} = c\mathbf{x}$ for some scalar c (which is the same thing as $\mathbf{x} = b\mathbf{y}$ for some scalar b: since $\mathbf{x}, \mathbf{y} \neq \mathbf{0}$ such a scalar multiplier in either case must be nonzero, so it can be brought to the other side as its reciprocal). In $\mathbf{R}^2$ and $\mathbf{R}^3$, this is a fact we can "see". In $\mathbf{R}^n$ for general n, this amounts to the second assertion in Theorem 2.3.2 below (for those who are interested). ■

Example 2.2.4 (Cosine similarity). Visualization in $\mathbf{R}^3$ suggest that for any n, two unit vectors in $\mathbf{R}^n$ or more generally two n-vectors $\mathbf{v}$ and $\mathbf{w}$ with the *same length* ℓ should be "close" when the angle θ between them is near 0° and should be "unrelated" when the angle between them is close to 90°. To turn this informal idea into precise mathematics, we go to the definitions and get a handle on $\|\mathbf{v} - \mathbf{w}\|$ by computing its square via the general formula $\mathbf{x} \cdot \mathbf{x} = \|\mathbf{x}\|^2$:

$$\begin{aligned}
\|\mathbf{v} - \mathbf{w}\|^2 = (\mathbf{v} - \mathbf{w}) \cdot (\mathbf{v} - \mathbf{w}) &= \mathbf{v} \cdot \mathbf{v} - \mathbf{v} \cdot \mathbf{w} - \mathbf{w} \cdot \mathbf{v} + \mathbf{w} \cdot \mathbf{w} \\
&= \|\mathbf{v}\|^2 - 2(\mathbf{v} \cdot \mathbf{w}) + \|\mathbf{w}\|^2 \\
&= \|\mathbf{v}\|^2 - 2\|\mathbf{v}\|\|\mathbf{w}\| \cos(\theta) + \|\mathbf{w}\|^2 \\
&= \ell^2 - 2\ell^2 \cos(\theta) + \ell^2 \\
&= 2\ell^2(1 - \cos(\theta)),
\end{aligned}$$

and this is very small for θ near 0° since then $1 - \cos(\theta) \approx 0$, so its square root $\|\mathbf{v} - \mathbf{w}\|$ is also small! This illustrates how vector algebra extracts geometric consequences for n-vectors with any n even though our literal visualization is limited to the case $n \leq 3$ (but thinking is seeing!).

Using the cosine of the angle as a measure of similarity for unit vectors in $\mathbf{R}^n$ for large n has a striking real-world application: the natural language processing technique called *latent semantic analysis* (also called *latent semantic indexing*). This is a powerful way to do information retrieval on the Internet and to search for "similar" documents; it avoids difficulties in linguistic-based methods caused by synonyms and multiple meanings of words. (Applying this to a large sample of papers in materials science was used in [**Tsh**] to predict that specific materials generate electricity from heat differences and so can improve fuel efficiency of cars, power smart watches via body heat, etc.).

The method assigns to every word from a collection of n documents an n-vector whose jth entry is a certain numerical statistic measuring the significance of the word in the jth document. (Most entries in each "word vector" are 0.) Sophisticated linear algebra methods at the end of this course (in Section 27.3) are applied to the resulting huge collection of "word vectors", yielding modified word vectors for which the cosine of angles between them is a useful measure of similarity among parts of documents. This is one of the techniques used by Spotify when it recommends songs and by the software DALLE-2 that creates striking images from text input, both using $\mathbf{R}^n$ with n in the millions. The use of cosine similarity to measure performance of large language models is mentioned here.

Since the method is based entirely on mathematics and *not* on any linguistic information at all (no dictionaries, no grammar, etc.), the technique is independent of the language in which documents are written and it is even effective at discovering similarities among documents written in *different* languages (even with different alphabets)! In fact, there's no need to limit oneself to comparing text in human languages: variants of the method have been applied with success to *biological problems* such as classifying proteins (by treating each long protein sequence as if it is a collection of "words", with the protein sequence then regarded as a "document"). ■

2.3. Pythagorean Theorem in $\mathbf{R}^n$ and the Cauchy–Schwarz Inequality. As an application of the dot product rules in Theorem 2.2.1, we can establish a version of the Pythagorean Theorem for n-vectors with any n (not just $n = 2$) and we can show that the subtlety lurking in the definition of "angle" between n-vectors (see Remark 2.1.14) is really not a problem at all.

Theorem 2.3.1 (Pythagoras). If n-vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are nonzero and perpendicular (i.e., at an angle of 90°) then

$$\|\mathbf{v}_1 + \mathbf{v}_2\|^2 = \|\mathbf{v}_1\|^2 + \|\mathbf{v}_2\|^2. \tag{2.3.1}$$

If we imagine a triangle with vertex at $\mathbf{0}$ and two sides along $\mathbf{v}_1$ and $\mathbf{v}_2$ then the other side corresponds to the displacement vector $\mathbf{v}_1 - \mathbf{v}_2$, so upon replacing $\mathbf{v}_2$ with its negative in (2.3.1) we see that this claimed equality really does express the geometric content of the Pythagorean Theorem.

PROOF. To establish (2.3.1), expand the left side as a dot product:

$$(\mathbf{v}_1 + \mathbf{v}_2) \cdot (\mathbf{v}_1 + \mathbf{v}_2) = \mathbf{v}_1 \cdot (\mathbf{v}_1 + \mathbf{v}_2) + \mathbf{v}_2 \cdot (\mathbf{v}_1 + \mathbf{v}_2) = \mathbf{v}_1 \cdot \mathbf{v}_1 + \mathbf{v}_1 \cdot \mathbf{v}_2 + \mathbf{v}_2 \cdot \mathbf{v}_1 + \mathbf{v}_2 \cdot \mathbf{v}_2.$$

We've used the rules for dot products at each step, to expand. But the common value $\mathbf{v}_1 \cdot \mathbf{v}_2$ and $\mathbf{v}_2 \cdot \mathbf{v}_1$ is 0 because $\mathbf{v}_1$ and $\mathbf{v}_2$ are assumed to be *perpendicular* (which means by definition that their dot product equals 0). Thus, the right side equals $\mathbf{v}_1 \cdot \mathbf{v}_1 + \mathbf{v}_2 \cdot \mathbf{v}_2 = \|\mathbf{v}_1\|^2 + \|\mathbf{v}_2\|^2$, as we wanted. ☐

Of course, this doesn't give a new proof of the Pythagorean Theorem in the Euclidean geometry case $n = 2$! Indeed, the *motivation for our definitions* of perpendicularity and more generally angle between vectors in $\mathbf{R}^n$ and length of vectors in $\mathbf{R}^n$ for general n (especially $n > 3$) came from our knowledge of how things work in $\mathbf{R}^2$ based on *knowing* the Pythagorean Theorem in plane geometry. Making up definitions of words cannot ever replace the work involved in proving a real theorem.

Since we now have several good properties of dot products in hand, we can establish a fact that we saw in Example 2.2.3 is needed to confirm that our definition of "angle" between nonzero n-vectors makes sense for any n:

Theorem 2.3.2 (Cauchy–Schwarz Inequality). For n-vectors $\mathbf{v}, \mathbf{w}$, we have

$$-\|\mathbf{v}\|\,\|\mathbf{w}\| \leq \mathbf{v} \cdot \mathbf{w} \leq \|\mathbf{v}\|\,\|\mathbf{w}\|$$

(or equivalently the absolute value $|\mathbf{v} \cdot \mathbf{w}|$ is at most $\|\mathbf{v}\|\,\|\mathbf{w}\|$). Moreover, one of the inequalities is an equality precisely when one of $\mathbf{v}$ or $\mathbf{w}$ is a scalar multiple of the other.

PROOF. If $\mathbf{v} = \mathbf{0}$ or $\mathbf{w} = \mathbf{0}$ then everything is clear (note that $\mathbf{0}$ is a scalar multiple of any n-vector: multiply it by the scalar 0), so now we assume $\mathbf{v}, \mathbf{w} \neq \mathbf{0}$. The idea of the proof is to explore how the length of $\mathbf{v} + x\mathbf{w}$ depends on x. This is most conveniently done by analyzing the squared-length, which is a dot product:

$$\|\mathbf{v} + x\mathbf{w}\|^2 = (\mathbf{v} + x\mathbf{w}) \cdot (\mathbf{v} + x\mathbf{w}).$$

Using Theorem 2.2.1(iii′) a few times, we have

$$\begin{aligned}(\mathbf{v} + x\mathbf{w}) \cdot (\mathbf{v} + x\mathbf{w}) &= (\mathbf{v} + x\mathbf{w}) \cdot \mathbf{v} + x((\mathbf{v} + x\mathbf{w}) \cdot \mathbf{w}) \\ &= \mathbf{v} \cdot \mathbf{v} + (x\mathbf{w}) \cdot \mathbf{v} + x((\mathbf{v} \cdot \mathbf{w}) + x(\mathbf{w} \cdot \mathbf{w})) \\ &= \mathbf{v} \cdot \mathbf{v} + x(\mathbf{w} \cdot \mathbf{v}) + x(\mathbf{v} \cdot \mathbf{w}) + x^2(\mathbf{w} \cdot \mathbf{w}).\end{aligned}$$

But $\mathbf{w} \cdot \mathbf{v} = \mathbf{v} \cdot \mathbf{w}$, so combining the middle two terms yields:

$$\|\mathbf{v} + x\mathbf{w}\|^2 = \|\mathbf{v}\|^2 + 2(\mathbf{v} \cdot \mathbf{w})x + \|\mathbf{w}\|^2 x^2.$$

The squared length of a vector is always ≥ 0, and it equals 0 precisely when the vector equals $\mathbf{0}$. But the vector $\mathbf{v} + x\mathbf{w}$ equals $\mathbf{0}$ for some value $x = c$ precisely when $\mathbf{v} = -c\mathbf{w}$, which is to say $\mathbf{v}$ is a scalar multiple of $\mathbf{w}$, and that is the same as $\mathbf{w}$ being a scalar multiple of $\mathbf{v}$ (since the scalar multiplier can be brought to the other side as its reciprocal as long as the scalar cannot be 0, and indeed such a scalar cannot be 0 since we have arranged that $\mathbf{v}, \mathbf{w} \neq \mathbf{0}$). So we just need to analyze what it means that the quadratic polynomial in x given by

$$q(x) = \|\mathbf{w}\|^2 x^2 + 2(\mathbf{v} \cdot \mathbf{w})x + \|\mathbf{v}\|^2$$

is always non-negative, and determine when this polynomial does actually attain the value 0 for some value of x.

Let's review when a quadratic polynomial $ax^2 + bx + c$ with *positive* leading coefficient (such as $a = \|\mathbf{w}\|^2$ for $q(x)$ above) is ≥ 0 everywhere. This happens precisely when its concave-up parabolic graph lies entirely on one side of the x-axis (possibly touching the x-axis at one point), which is exactly the situation that the graph does *not* cross the x-axis at two different points. This is exactly the situation when the output of the quadratic formula does *not* yield two different real numbers. The opposite case of *having two different real roots* occurs exactly when the "$b^2 - 4ac$" part of the quadratic formula inside the square-root is > 0, so in our situation we must have the exactly opposite situation: $b^2 - 4ac \leq 0$, with equality happening precisely when there is a real root.

Applying the preceding review with $a = \|\mathbf{w}\|^2$, $b = 2(\mathbf{v} \cdot \mathbf{w})$, $c = \|\mathbf{v}\|^2$ for $q(x)$, we get

$$(2(\mathbf{v} \cdot \mathbf{w}))^2 - 4\|\mathbf{w}\|^2\|\mathbf{v}\|^2 \leq 0$$

with equality happening exactly when $\mathbf{v}$ and $\mathbf{w}$ are scalar multiples of each other. Bringing the second term on the left over to the other side, we conclude that

$$(2(\mathbf{v} \cdot \mathbf{w}))^2 \leq 4\|\mathbf{w}\|^2\|\mathbf{v}\|^2$$

with equality precisely when $\mathbf{v}$ and $\mathbf{w}$ are scalar multiples of each other. Dividing each side by 4, this is the same as the inequality

$$|\mathbf{v} \cdot \mathbf{w}|^2 \leq (\|\mathbf{w}\| \, \|\mathbf{v}\|)^2,$$

so taking square roots of both sides gives what we want. ☐

2.4. The correlation coefficient. Given data points $(x_1, y_1), \ldots, (x_n, y_n)$, it is often useful to seek a line which gives a "best fit" to this collection of points. The first example of this arose in the early 19th century when astronomers lost an asteroid that they had recently discovered and sought to rediscover it via extrapolation from the limited measurements they had made (we will return to this in Example 7.2.1).

The problem of finding a "best fit" line to some data is called *linear regression*, and we will address that task using later linear algebra techniques in Chapter 7. But at a more basic level we may seek a measure of the extent to which it is *reasonable* to try to find a line that could be regarded as a good fit to the data (setting aside what that specific line may be). There is a widely used measure of whether one should seek such a line: this measure is called the *correlation coefficient* of the data points.

Example 2.4.1. Consider the 5 data points (x_i, y_i) given by

$$(-3, 4), \ \ (-2, 1), \ \ (0, -1), \ \ (1, -1), \ \ (4, -3).$$

These points and the corresponding (not so good) "line of best fit" are shown in Figure 2.4.1 below. In Chapter 7 we will learn how to determine which line may be reasonably considered to be the one that best fits the data, but it is appropriate to *first* ask if such a line should be considered useful or not.

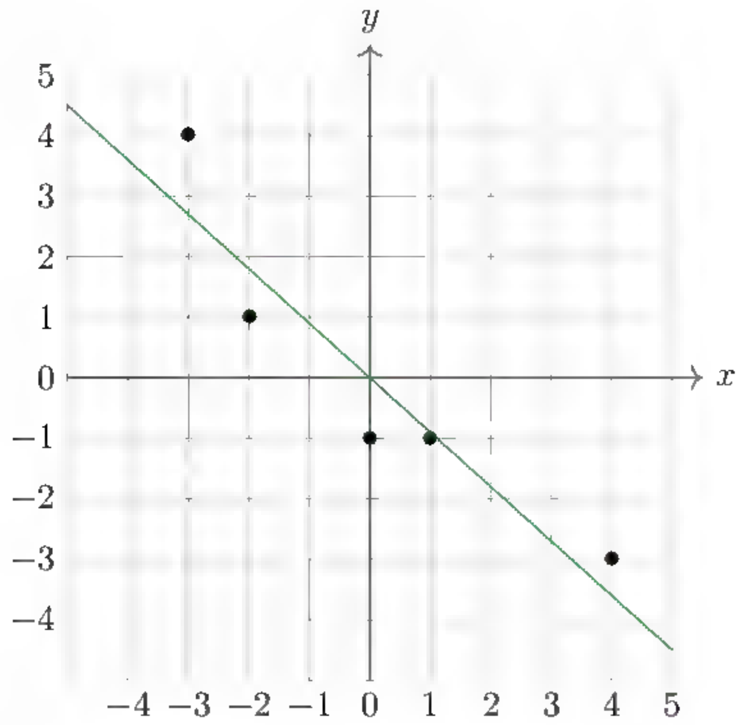


FIGURE 2.4.1. Data points: $(-3,4)$, $(-2,1)$, $(0,-1)$, $(1,-1)$, $(4,-3)$.

■

In general, to what extent is there an approximately "linear" relationship between y_i and x_i for n given data points (x_i, y_i)? *Even though this question concerns points in the plane* $\mathbf{R}^2$*, to answer it we will use the language of vectors in* $\mathbf{R}^n$! This illustrates how considerations with $\mathbf{R}^n$ for $n > 3$ arise very naturally in contexts that do not initially seem to involve anything outside $\mathbf{R}^2$ or $\mathbf{R}^3$.

Let's describe a process that answers this question for the data in Example 2.4.1, using the 5-vectors

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_5 \end{bmatrix} = \begin{bmatrix} -3 \\ -2 \\ 0 \\ 1 \\ 4 \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_5 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ -1 \\ -1 \\ -3 \end{bmatrix}$$

whose entries respectively record the x-coordinates and the y-coordinates of the data. The general procedure will involve working with the associated *unit vectors* $\mathbf{X}/\|\mathbf{X}\|$ and $\mathbf{Y}/\|\mathbf{Y}\|$ obtained by dividing each of the 5-vectors by its length. Since any effect of multiplying all entries of $\mathbf{X}$ or $\mathbf{Y}$ by a common scaling factor, such as arise under change of units of measurement (e.g., measuring in feet or in meters), cancels out when passing to these unit vectors, we get a concept that is insensitive to "change of units" in the measurements and hence has more genuine significance.

Consider n data points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ in $\mathbf{R}^2$. Assume (as will happen in examples you are given) they don't all lie on a common vertical line nor on a common horizontal line (i.e., the x_i's are not all equal to each other, and the y_i's are not all equal to each other, so in particular $\mathbf{X}, \mathbf{Y} \neq \mathbf{0}$).

Definition 2.4.2. In the above setup, assume furthermore (as will happen in examples you are given) that the averages $\overline{x} = (1/n)\sum x_i$ and $\overline{y} = (1/n)\sum y_i$ of the x-coordinates and of the y-coordinates both equal 0. The *correlation coefficient* r between the x_i's and y_i's is defined to be the cosine of the angle between $\mathbf{X}$ and $\mathbf{Y}$, or equivalently between the unit vectors $\mathbf{X}/\|\mathbf{X}\|$ and $\mathbf{Y}/\|\mathbf{Y}\|$:

$$r = \text{cosine of the angle between } \mathbf{X} \text{ and } \mathbf{Y} = \frac{\mathbf{X}\cdot\mathbf{Y}}{\|\mathbf{X}\|\|\mathbf{Y}\|} = \frac{\mathbf{X}}{\|\mathbf{X}\|}\cdot\frac{\mathbf{Y}}{\|\mathbf{Y}\|}. \qquad (2.4.1)$$

Example 2.4.3. Returning to Example 2.4.1, the respective lengths of the corresponding 5-vectors are $\|\mathbf{X}\| = \sqrt{30}$, $\|\mathbf{Y}\| = \sqrt{28}$. Thus, the correlation coefficient is

$$r = \frac{\mathbf{X} \cdot \mathbf{Y}}{\|\mathbf{X}\| \|\mathbf{Y}\|} = \frac{-27}{\sqrt{30 \cdot 28}} \approx -0.9316.$$

We will come back to the meaning of this in Example 2.4.6. ■

Remark 2.4.4. You may be bothered by the assumption that the averages $\overline{x}$ and $\overline{y}$ of the coordinates of the data both equal 0, since in practice it is rarely satisfied. What is done in real-world problems is that the data is *recentered*: we replace x_i with $\widehat{x}_i = x_i - \overline{x}$ and replacing y_i with $\widehat{y}_i = y_i - \overline{y}$. Such subtraction of the averages makes "center of mass" move to $(0, 0)$ (i.e., $\overline{\widehat{x}}, \overline{\widehat{y}} = 0$). The correlation coefficient of the original data is *defined* to be the application of Definition 2.4.2 to this recentered data. You will learn more about such recentering in a course on statistics; we won't dwell on it here.

Having defined the correlation coefficient as a cosine and computed it in an example, we next address the meaning of this number.

Theorem 2.4.5. The correlation r always lies between -1 and 1. When r is close to 1 this means the data points (x_i, y_i) are close to a line of positive slope, and when r is close to 1 this means that the data points (x_i, y_i) are close to a line of negative slope.

A correlation coefficient close to 0 means that there *does not* appear to be a strong linear relation between x_i and y_i.

A general explanation of Theorem 2.4.5 is best expressed in the language of vector algebra, so we postpone it (for those who are interested) to Section 7.5. But we can get some insight into why correlation coefficients measure how close the y-components of the data are to being linearly related to the x-components by considering some special situations as follows.

Let's imagine the case of 3 data points $(x_1, y_1), (x_2, y_2)$, and (x_3, y_3) for which the y-components depend *exactly* linearly on the x-components. In other words, suppose we have exact relationships

$$y_i = mx_i + b$$

for some constants m and b with $m \neq 0$. Assume furthermore (as in the setup for Definition 2.4.2) that the x_i's aren't all equal to each other, the y_i's aren't all equal to each other (so $\mathbf{X}, \mathbf{Y} \neq \mathbf{0}$), and that the averages $\overline{x} = \frac{1}{3}(x_1 + x_2 + x_3)$ and $\overline{y} = \frac{1}{3}(y_1 + y_2 + y_3)$ both equal 0. Since

$$\overline{y} = \frac{1}{3}(y_1 + y_2 + y_3) = \frac{1}{3}((mx_1 + b) + (mx_2 + b) + (mx_3 + b)) = m\overline{x} + b$$

with $\overline{x} = 0$ and $\overline{y} = 0$, we must have $b = 0$. Hence, $y_i = mx_i$ for all i, so $\mathbf{Y} = m\mathbf{X}$.

So in this case $\mathbf{Y}$ is m times $\mathbf{X}$. Hence, the nonzero vectors $\mathbf{Y}$ and $\mathbf{X}$ point in the same direction if $m > 0$ and point in *opposite* directions if $m < 0$. In other words the angle between $\mathbf{X}$ and $\mathbf{Y}$ is $0°$ if the slope m is positive and is 180° if the slope m is negative (see Example 2.2.3). The correlation coefficient is the cosine of the angle between these vectors by its definition, so it is $\cos(0°) = 1$ if $m > 0$ and is $\cos(180°) = -1$ if $m < 0$.

If one has data $(x_1, y_1), \ldots, (x_n, y_n)$ that lie near a line of slope $m \neq 0$, then the correlation coefficient r reflects this by being very near 1 if $m > 0$, and very near -1 if $m < 0$. If, however, the data is scattered randomly in the plane, and not really near any line, then one would expect the correlation coefficient to be near zero.

Often people work with r^2, which is always non-negative. This is

$$r^2 = \frac{(\mathbf{X} \cdot \mathbf{Y})^2}{\|\mathbf{X}\|^2 \|\mathbf{Y}\|^2}; \tag{2.4.2}$$

it is near 0 when there is little correlation, and near 1 when there's a strong linear relationship (without specifying the sign of the slope: r may be near 1 or near -1).

Don't confuse the value of r with the slope of a "best-fit line"! The nearness of r^2 to 1 (or of r to ± 1) is a measure of quality of fit. The actual slope of the best-fit line (which could be any real number at all) has nothing whatsoever to do with the value of r (which is always between -1 and 1).

Example 2.4.6. In Example 2.4.3, we found that $r \approx -0.9316$. This is very close to -1, so the data should be near a line with negative slope (though not slope -1!). This property can be seen from the plot of the 5 given data points in Figure 2.4.1. ■

If you look at any work that tries to *quantify* the extent of the correlation between, well, practically anything (supply and demand? attendance and grades? air temperature and the running speed of lizards? boredom and Internet addiction?) you will find that the correlation coefficient is an essential part of the analysis. Correlation coefficients go hand in hand with linear regression (finding a "best fit" line for data) and help one to understand how *meaningful* the results of a linear regression are.

But the correlation coefficient is just a single number and so cannot be ascribed magical powers: it is a useful way to quantify the informal idea that data lies "near" a line but one should always look at the data plot to make sure nothing strange is happening. Moreover, in the spirit of the old adage that "correlation does not imply causation" (e.g., monthly crime rates and monthly ice cream sales), note that the correlation coefficient treats $\mathbf{X}$ and $\mathbf{Y}$ in a symmetric manner whereas any causal relationship is asymmetric.

Example 2.4.7. Here is an example of how correlation coefficients are used in analyzing the statistics in a lab science experiment. We consider an example from neuroscience. In this discipline, correlation coefficients are often used to understand the relationship between the activity level of a neuron or group of neurons and observations by the subject of the experiment (such as a monkey, dog, or human).

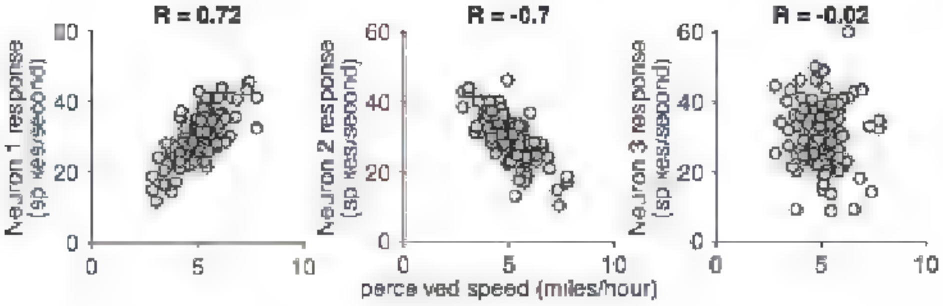
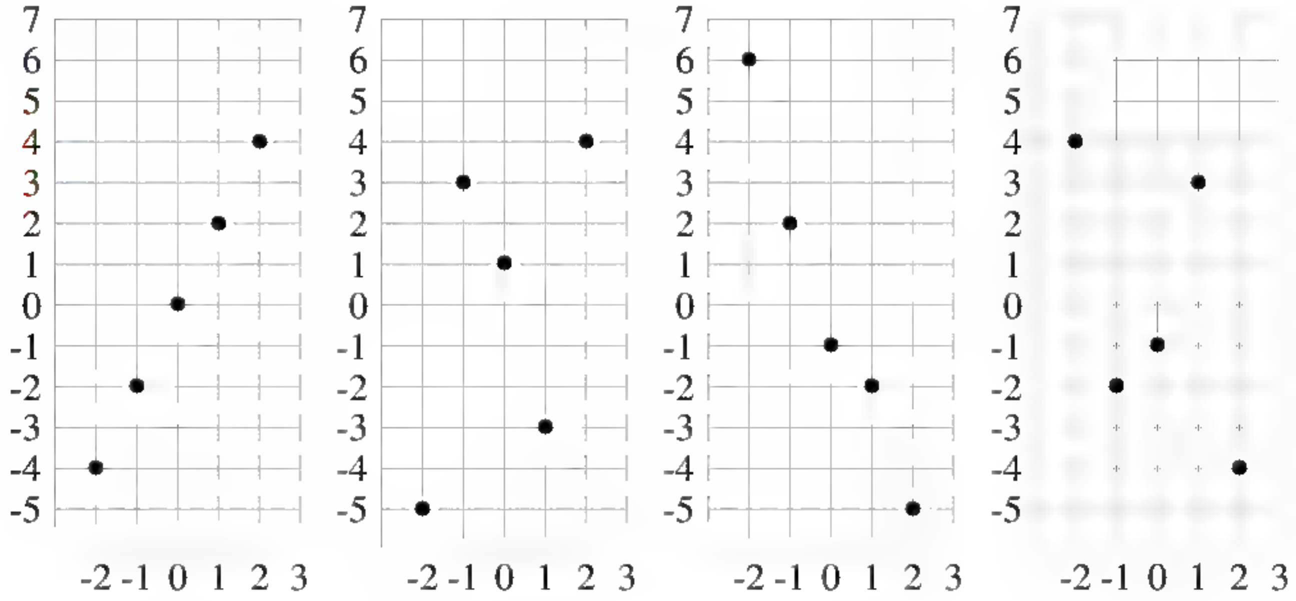


FIGURE 2.4.2. Activity graphs for three neurons in response to a visual stimulus

Imagine showing a subject a moving object. Neurons in the visual cortex respond to moving stimuli by firing action potentials, or spikes, which are changes in voltage that are used to communicate with other neurons in the network. We can calculate a correlation coefficient between a neuron's activity (quantified as the number of spikes it fires per second) and the speed of the moving object. Figure 2.4.2 shows such data for three stimulated neurons.

The first neuron is selective for fast speeds: it fires more when the stimulus is moving quickly, so its activity is positively correlated with the speed. The second neuron is selective for slow speeds, so its activity is negatively correlated with the speed. The response of the final neuron is uncorrelated with the speed, which might mean that this neuron has a role in the brain that is unrelated to observing motion. ■

Example 2.4.8. For each of the following collections of 5 data points, all of which are plotted in Figure 2.4.3 below, verify the correlation coefficient and inspect the plot to see if it lies near a line.



$r = 1$ $\quad$ $r \approx .490$ $\quad$ $r \approx -.983$ $\quad$ $r \approx -.513$

FIGURE 2.4.3. Plots for Example 2.4.8

- $(-2,-4),(-1,-2),(0,0),(1,2),(2,4)$ with $r = 1$.
- $(-2,-5),(-1,3),(0,1),(1,-3),(2,4)$ with $r = \sqrt{6}/5 \approx 0.490$.
- $(-2,6),(-1,2),(0,-1),(1,-2),(2,-5)$ with $r = -13/(5\sqrt{7}) \approx -0.983$.
- $(-2,4),(-1,-2),(0,-1),(1,3),(2,-4)$ with $r = -11/(2\sqrt{115}) \approx -0.513$.

(In each case, we have arranged that the averages $\overline{x}$ and $\overline{y}$ equal 0.) ■

Remark 2.4.9 (optional). Let's see why the correlation coefficient equals 1 precisely when the points (x_i, y_i) all lie *exactly* on a line $y = mx$ whose slope m is positive. (We assume as always that the data doesn't all lie on a common vertical line or a common horizontal line, and that the averages $\overline{x}$ and $\overline{y}$ equal 0.) By then replacing y_i with $-y_i$ everywhere, it would follow that the correlation coefficient equals -1 precisely when the points (x_i, y_i) all lie *exactly* on a line $y = mx$ whose slope m is negative.

Note that $\mathbf{X}, \mathbf{Y} \neq \mathbf{0}$ since we assumed the data points aren't on a common horizontal line and aren't on a common vertical line. We want to show that the correlation coefficient is 1 precisely when the $\mathbf{Y} = m\mathbf{X}$ for some $m > 0$.

But the correlation coefficient is the cosine of the angle between the nonzero vectors $\mathbf{X}$ and $\mathbf{Y}$, so the correlation coefficient is equal to 1 precisely when the angle between $\mathbf{X}$ and $\mathbf{Y}$ is 0°. In Example 2.2.3 we discussed why the angle θ between the (nonzero) vectors $\mathbf{X}$ and $\mathbf{Y}$ is 0° precisely when $\mathbf{Y} = m\mathbf{X}$ for some $m > 0$.

2.5. Dot product formula for angles in $\mathbf{R}^3$. In this section, we derive the formula (2.1.3) for the angle between two nonzero vectors in $\mathbf{R}^3$, and we use the Cauchy–Schwarz Inequality (Theorem 2.3.2) to establish the "triangle inequality" from Remark 1.6.12. First, let's establish (2.1.3).

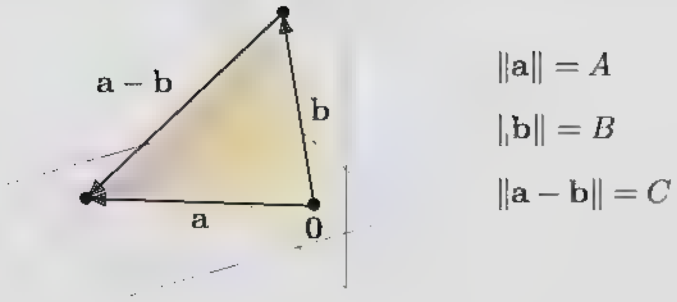


FIGURE 2.5.1. Geometric setup for dot product formula for angle between two vectors

Consider a triangle in $\mathbf{R}^3$ as in Figure 2.5.1, with side lengths $A = \|\mathbf{a}\|, B = \|\mathbf{b}\|$ and $C = \|\mathbf{a} - \mathbf{b}\|$. We want to compute the angle θ between the sides along $\mathbf{a}$ and $\mathbf{b}$. The Law of Cosines says that $C^2 = A^2 + B^2 - 2AB\cos(\theta)$. Solving for $\cos(\theta)$, this says

$$\cos(\theta) = \frac{A^2 + B^2 - C^2}{2AB}. \tag{2.5.1}$$

We have $A^2 = \|\mathbf{a}\|^2 = a_1^2 + a_2^2 + a_3^2$ and $B^2 = \|\mathbf{b}\|^2 = b_1^2 + b_2^2 + b_3^2$ by the 3-dimensional version of the Pythagorean Theorem (see Example 1.6.5), and likewise

$$C^2 = \|\mathbf{a} - \mathbf{b}\|^2 = (a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2.$$

Expanding out each square and simplifying, there is a lot of cancellation and one gets

$$A^2 + B^2 - C^2 = 2(a_1b_1 + a_2b_2 + a_3b_3) = 2(\mathbf{a} \cdot \mathbf{b}).$$

Note the appearance of the dot product! Plugging this back into (2.5.1) and cancelling the common factor of 2 in the numerator and denominator thereby establishes (2.1.3).

To give an interesting application of the properties of dot products, let's next revisit the Cauchy–Schwarz Inequality

$$-1 \leq \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} \leq 1$$

from Theorem 2.3.2 and show that this provides *exactly* what is needed to answer the mystery raised in Remark 1.6.12 concerning whether or not a "triangle inequality" (1.6.1) holds in $\mathbf{R}^n$ for any n. In Remark 1.6.12 we saw that (1.6.1) amounts to determining if

$$\|\mathbf{v} + \mathbf{w}\| \overset{?}{\leq} \|\mathbf{v}\| + \|\mathbf{w}\| \tag{2.5.2}$$

for all n-vectors $\mathbf{v}, \mathbf{w}$ (for all $n \geq 1$).

For $n \leq 3$, (2.5.2) can be visualized by drawing suitable triangles, and our aim here is to establish it for all n by a uniform argument by using the Cauchy–Schwarz Inequality. This illustrates the power of our "geometric intuition" as a guide for what to expect to be true when thinking about length and distance in $\mathbf{R}^n$ regardless of how big n may be.

Theorem 2.5.1 (Triangle Inequality). For any n-vectors $\mathbf{v}, \mathbf{w}$, the inequality (2.5.2) holds.

PROOF. For any $a, b \geq 0$, the inequality $a \leq b$ holds precisely when $a^2 \leq b^2$ (this is not true if we don't require non-negativity: $-3 \leq 2$ but $(-3)^2 = 9 > 4 = 2^2$). Thus, to determine if (2.5.2) always holds it suffices to check after squaring both sides. But the squared length of a vector is its dot product against itself, so squaring both sides turns our task into determining whether or not the inequality $(\mathbf{v} + \mathbf{w}) \cdot (\mathbf{v} + \mathbf{w}) \leq \|\mathbf{v}\|^2 + 2\|\mathbf{v}\|\|\mathbf{w}\| + \|\mathbf{w}\|^2$ holds for all n-vectors $\mathbf{v}, \mathbf{w}$.

By the general algebraic rules for dot products, the left side is equal to

$$\mathbf{v}\cdot\mathbf{v}+\mathbf{v}\cdot\mathbf{w}+\mathbf{w}\cdot\mathbf{v}+\mathbf{w}\cdot\mathbf{w}=\|\mathbf{v}\|^2+2(\mathbf{v}\cdot\mathbf{w})+\|\mathbf{w}\|^2,$$

so our task is the same as showing $\|\mathbf{v}\|^2+2(\mathbf{v}\cdot\mathbf{w})+\|\mathbf{w}\|^2 \le \|\mathbf{v}\|^2+2\|\mathbf{v}\|\,\|\mathbf{w}\|+\|\mathbf{w}\|^2$. Cancelling the common occurrence of $\|\mathbf{v}\|^2$ and $\|\mathbf{w}\|^2$ on both sides turns this into checking the inequality

$$2(\mathbf{v}\cdot\mathbf{w})\le 2\|\mathbf{v}\|\|\mathbf{w}\|,$$

or equivalently

$$\mathbf{v}\cdot\mathbf{w}\le\|\mathbf{v}\|\|\mathbf{w}\|$$

for all n-vectors $\mathbf{v}$ and $\mathbf{w}$. But this is part of the Cauchy–Schwarz Inequality, so we are done! $\square$

Remark 2.5.2. A neat application of (2.1.3) is to combine it with dot product algebra in Theorem 2.2.1 to establish the Law of Sines, as we now explain. The Law of Sines says that for any triangle in $\mathbf{R}^3$ (or $\mathbf{R}^2$), if P is a vertex at which the interior angle is θ and the length of opposite side is L then the ratio $(\sin\theta)/L$ is the *same* regardless of which P we chose. These ratios are all positive (since $\sin\theta>0$ for $0^\circ<\theta<180^\circ$), so to show that such ratios are the same for all three choices of vertex P it is equivalent to show that the squares of these ratios are the same.

Our proof will be illuminating in a different way from most proofs since it explains the equality of the ratios "$(\sin\theta)/L$" for all three vertices via a formula involving all three sides *treated on equal footing* (another such proof is due to Ptolemy of Alexandria [KS, Figure 1, Prop. 1(i), Cor. 2(i)]):

$$\left(\frac{\sin\theta}{L}\right)^2=\frac{(\mathbf{x}\cdot\mathbf{y})(\mathbf{x}\cdot\mathbf{z})+(\mathbf{y}\cdot\mathbf{x})(\mathbf{y}\cdot\mathbf{z})+(\mathbf{z}\cdot\mathbf{x})(\mathbf{z}\cdot\mathbf{y})}{(\mathbf{x}\cdot\mathbf{x})(\mathbf{y}\cdot\mathbf{y})(\mathbf{z}\cdot\mathbf{z})} \tag{2.5.3}$$

for vectors $\mathbf{x},\mathbf{y},\mathbf{z}$ along the three sides directed so $\mathbf{x}+\mathbf{y}+\mathbf{z}=\mathbf{0}$ (i.e., the directions point "one way" around the triangle; which way doesn't matter, since negating all three has no effect on the formula), as illustrated in Figure 2.5.2 below. The denominator on the right side of (2.5.3) is unaffected by rearranging the sides; the numerator is also unaffected since it is a sum of three terms, each of which is the product of the dot products of one vector against the other two. We are going to prove (2.5.3).

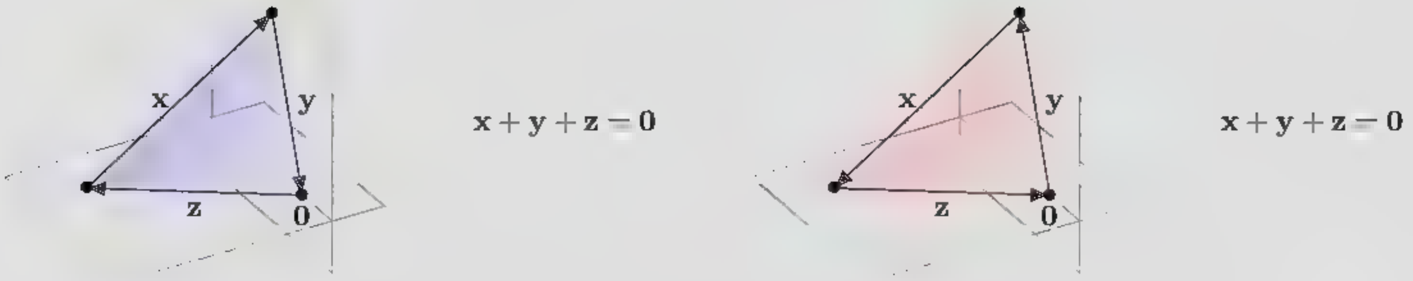


FIGURE 2.5.2. Sides of a triangle directed as vectors whose sum is $\mathbf{0}$

PROOF. Move the triangle so that one of the vertices is at $\mathbf{0}$; it looks as in Figure 2.5.1. Let θ be the interior angle at the vertex $\mathbf{0}$, so $\cos\theta=(\mathbf{a}\cdot\mathbf{b})/(\|\mathbf{a}\|\|\mathbf{b}\|)$ by (2.1.3) and $L=\|\mathbf{a}-\mathbf{b}\|$ by definition (implicitly using the parallelogram law to interpret $\mathbf{a}-\mathbf{b}$ geometrically). Hence,

$$\left(\frac{\sin\theta}{L}\right)^2=\frac{(\sin\theta)^2}{L^2}=\frac{1-(\cos\theta)^2}{\|\mathbf{a}-\mathbf{b}\|^2}=\frac{1-((\mathbf{a}\cdot\mathbf{b})/(\|\mathbf{a}\|\,\|\mathbf{b}\|))^2}{\|\mathbf{a}-\mathbf{b}\|^2}=\frac{\|\mathbf{a}\|^2\|\mathbf{b}\|^2-(\mathbf{a}\cdot\mathbf{b})^2}{(\|\mathbf{a}-\mathbf{b}\|\|\mathbf{a}\|\|\mathbf{b}\|)^2}.$$

We have arrived at an expression $f(\mathbf{a},\mathbf{b},\mathbf{a}-\mathbf{b})$ in terms of the three sides of the triangle $\mathbf{a},\mathbf{b},\mathbf{a}-\mathbf{b}$. The way each side appears in the expression as a vector is insensitive to negating it (which is good: the choice of "direction" along each side should not matter).

If we had focused on the squared ratio associated to either of the other two vertices then we would obtain a modified version of the same expression in which the roles of the three sides would be shuffled around (up to a choice of direction along each, which we have noted doesn't matter in the end): we'd get $f(\mathbf{b}, \mathbf{a}-\mathbf{b}, \mathbf{a})$ and $f(\mathbf{a}, \mathbf{a}-\mathbf{b}, \mathbf{b})$ (again keep in mind that f is insensitive to negating any of those three side directions). Our goal is to show that all three of these expressions are equal to each other.

Since $f(\mathbf{a}, \mathbf{b}, \mathbf{a}-\mathbf{b}) = f(-\mathbf{a}, \mathbf{b}, \mathbf{a}-\mathbf{b})$ and the three vectors $-\mathbf{a}, \mathbf{b}, \mathbf{a}-\mathbf{b}$ sum to $\mathbf{0}$, our task reduces to a more "symmetric" goal: if $\mathbf{x}, \mathbf{y}, \mathbf{z}$ are 3-vectors that sum to $\mathbf{0}$, then we claim that the ratio

$$f(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \frac{\|\mathbf{x}\|^2\|\mathbf{y}\|^2 - (\mathbf{x}\cdot\mathbf{y})^2}{(\|\mathbf{z}\|\|\mathbf{x}\|\|\mathbf{y}\|)^2} = \frac{(\mathbf{x}\cdot\mathbf{x})(\mathbf{y}\cdot\mathbf{y}) - (\mathbf{x}\cdot\mathbf{y})^2}{(\|\mathbf{z}\|\|\mathbf{x}\|\|\mathbf{y}\|)^2}$$

is *unaffected* by rearranging the three vectors (i.e., $f(\mathbf{x}, \mathbf{y}, \mathbf{z}) = f(\mathbf{y}, \mathbf{z}, \mathbf{x}) = f(\mathbf{x}, \mathbf{z}, \mathbf{y}) = \cdots$). In fact, this statement is meaningful with n-vectors for any n (there is nothing special about 3-vectors) and we claim it holds in that generality. The denominator is visibly unaffected by such rearrangement, and for the numerator we use an identity: under the hypothesis $\mathbf{x}+\mathbf{y}+\mathbf{z} = \mathbf{0}$ that treats all three of the vectors on equal footing, we claim an equality with an expression that is visibly unaffected by rearrangement:

$$(\mathbf{x}\cdot\mathbf{x})(\mathbf{y}\cdot\mathbf{y}) - (\mathbf{x}\cdot\mathbf{y})^2 = (\mathbf{x}\cdot\mathbf{y})(\mathbf{x}\cdot\mathbf{z}) + (\mathbf{y}\cdot\mathbf{x})(\mathbf{y}\cdot\mathbf{z}) + (\mathbf{z}\cdot\mathbf{x})(\mathbf{z}\cdot\mathbf{y}). \tag{2.5.4}$$

Why is the identity (2.5.4) true? If you plug $\mathbf{z} = -\mathbf{x}-\mathbf{y}$ into the right side and carefully expand out the dot products in accordance with Theorem 2.2.1, you'll get a big expression involving products of dot products among $\mathbf{x}$ and $\mathbf{y}$. Upon close inspection there is a huge amount of cancellation, leaving the desired left side of (2.5.4) after the dust settles. Voilà. □

Chapter 2 highlights (links to highlights in [illegible] and next chapters)

Notation	**Meaning**	**Location in text**
$\mathbf{x} \cdot \mathbf{y}$	dot product of n-vectors $\mathbf{x}$ and $\mathbf{y}$	Def. 2.1.6(i)
$\overline{v}$	for n-vector $\begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$, it is the average $(1/n)\sum_{j=1}^{n} v_j$ of all entries	Def. 2.4.2
r	correlation coefficient for n data points not on a common vertical or horizontal line (and with each coordinate averaging to 0)	(2.4.1)

Concept	**Meaning**	**Location in text**
dot product $\mathbf{x} \cdot \mathbf{y}$	operation on n-vectors $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$, yielding $x_1 y_1 + \cdots + x_n y_n$	Definition 2.1.6(i)
angle θ	for nonzero n-vectors $\mathbf{x}$ and $\mathbf{y}$, defined by $\cos(\theta) = (\mathbf{x} \cdot \mathbf{y})/(\lVert\mathbf{x}\rVert \lVert\mathbf{y}\rVert)$ with $0^\circ \leq \theta \leq 180^\circ$	Definition 2.1.6(ii)
orthogonal (or perpendicular)	a pair of n-vectors whose dot product equals 0	Definition 2.1.6(iii)
correlation coefficient r	for nonzero $\mathbf{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ and $\mathbf{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$ arising from n data points (x_i, y_i) with $\overline{x} = 0$ and $\overline{y} = 0$, it is cosine of angle between $\mathbf{X}$ and $\mathbf{Y}$	(2.4.1)

Result	**Meaning**	**Location in text**
cosine formula	for nonzero 2-vectors or 3-vectors $\mathbf{a}$ and $\mathbf{b}$, angle θ between them satisfies $\cos(\theta) = (\mathbf{a} \cdot \mathbf{b})/(\lVert\mathbf{a}\rVert \lVert\mathbf{b}\rVert)$	Theorems 2.1.1, 2.1.3
length via dot product	$\lVert\mathbf{v}\rVert = \sqrt{\mathbf{v} \cdot \mathbf{v}}$ for an n-vector $\mathbf{v}$	Theorem 2.2.1(ii)
properties of dot products	dot products satisfy analogues of properties of multiplication of ordinary numbers, including interaction with vector addition and scalar multiplication	Theorem 2.2.1(i),(iii),(iii′)
$-1 \leq r \leq 1$	for data points in $\mathbf{R}^2$, correlation coefficient r lies between -1 and 1	Theorem 2.4.5
interpretation of $r \approx \pm 1$ and $r \approx 0$	data points in $\mathbf{R}^2$ are well-approximated by some line if $r \approx \pm 1$ (don't confuse r with slope!), poorly approximated by all lines if $r \approx 0$	discussion near (2.4.2)

Skill	**Location in text**
compute dot products of n-vectors (any n)	Example 2.1.7
express angle between nonzero n-vectors in terms of dot products, relate this to an angle between lines when $n = 2, 3$	Examples 2.1.5, 2.1.13
verify orthogonality for two n-vectors (any n)	Example 2.1.2
compute correlation coefficient for n given data points in $\mathbf{R}^2$ using formula with dot products and lengths of associated n-vectors	Examples 2.4.3, 2.4.8

3. Planes in $\mathbf{R}^3$

As a geometric application of vector algebra, we now use the viewpoint of vectors and dot products to study geometric questions about planes in familiar 3-dimensional space.

By the end of this chapter, you should be able to:

- transition between different descriptions of a plane in $\mathbf{R}^3$ using vector algebra (e.g., given three points on a plane not on a common line, you should be able to produce: an equation for the plane, a normal vector to the plane, and a parametrization of the plane);
- use the equational form to determine if points lie on the same or different sides of a plane;
- interpret the parametric form in terms of displacement vectors from a point on the plane.

3.1. Line of sight in computer graphics. Suppose we are making a video game and need to figure out if the roof of a house blocks the view of a tree in the backyard from someone standing at the bottom of the driveway. How can we analyze this? A version of this type of problem arises all the time in computer graphics: a computer rendering a scene must know when one object blocks another from view.

Imagine the relevant part of the roof is flat (so it lies in a single plane, perhaps tilted at some angle relative to the ground). A simplified mathematical model of this question (ignoring more refined geometric information concerning actual distances along the roof) is then this:

> **Question**: Given points P and Q and a plane in $\mathbf{R}^3$, do P and Q lie on the *same* side of the plane (in which case the plane doesn't obstruct P's view of Q) or on *opposite* sides?

(See Figure 3.1.6 for an illustration of several points on either side of a plane, or imagine it in your head.)

An answer to this Question would determine if the viewer's line of sight and points of the tree are on the same side of the plane containing the relevant flat part of the roof. If everything is on the same side of that plane, there is no obstruction. (Imagine the roof is very large, so edge effects can be ignored.) To answer the Question, we now show how to describe a plane in $\mathbf{R}^3$ using an equation generalizing the one we know describes a line in $\mathbf{R}^2$.

Let's first review the equation to describe a line in $\mathbf{R}^2$. The collection of points (x, y) in $\mathbf{R}^2$ satisfying $5x - 7y = 3$ is a line in $\mathbf{R}^2$ since the equation can be rewritten as $y = (5/7)x - 3/7$, which we recognize as the line with slope $5/7$ passing through $(0, -3/7)$ on the y-axis. More generally, the solutions (x, y) to an equation of the form $ax + by = c$, with at least one of the constants a or b nonzero, is a line in $\mathbf{R}^2$. Indeed, if $b \neq 0$ then we can divide both sides by b to rewrite it as $y = -(a/b)x + c/b$ (which we recognize as a "slope-intercept form") and if $b = 0$ (so $a \neq 0$) then we can divide by a to rewrite it as $x = c/a$ (a vertical line: infinite slope). A merit of writing the equation of a line in $\mathbf{R}^2$ in the form $ax + by = c$ is that this treats x and y on equal footing.

The generalization to planes in $\mathbf{R}^3$ looks almost the same!

The collection of points (x, y, z) in $\mathbf{R}^3$ satisfying $3x - 4y + 2z = 10$ is a plane, and more generally the collection of points (x, y, z) in $\mathbf{R}^3$ satisfying an equation of the form

$$ax + by + cz = d,$$

with at least one of the constants a, b, or c nonzero, is a *plane* in $\mathbf{R}^3$. In particular, although the equation $x = 0$ on $\mathbf{R}^2$ defines a line (the y-axis, consisting of points $(0, y)$), the "same" equation $x = 0$ on $\mathbf{R}^3$ defines a *plane*, namely the vertical yz-plane consisting of points $(0, y, z)$.

One good way to visualize why an equation of this type really does describe a plane is given in Section 3.4. There are actually multiple completely equivalent ways to describe a plane. We first give 2 ways to describe the *same* plane in $\mathbf{R}^3$ (it may not be apparent to you yet that the methods do both describe the same plane – we discuss this further in Section 3.2). Then we'll give 2 additional methods.

(a) (equational form) The set of all points (x, y, z) in $\mathbf{R}^3$ which are solutions to $x + 2y + 3z = 4$; see Figure 3.1.1. This is a special case of the preceding discussion.

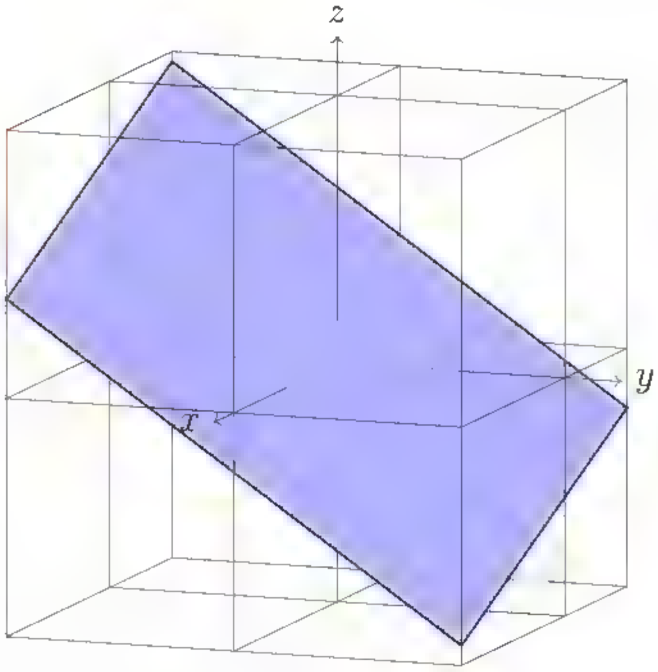


FIGURE 3.1.1. (a) A plane given by the equation $x + 2y + 3z = 4$

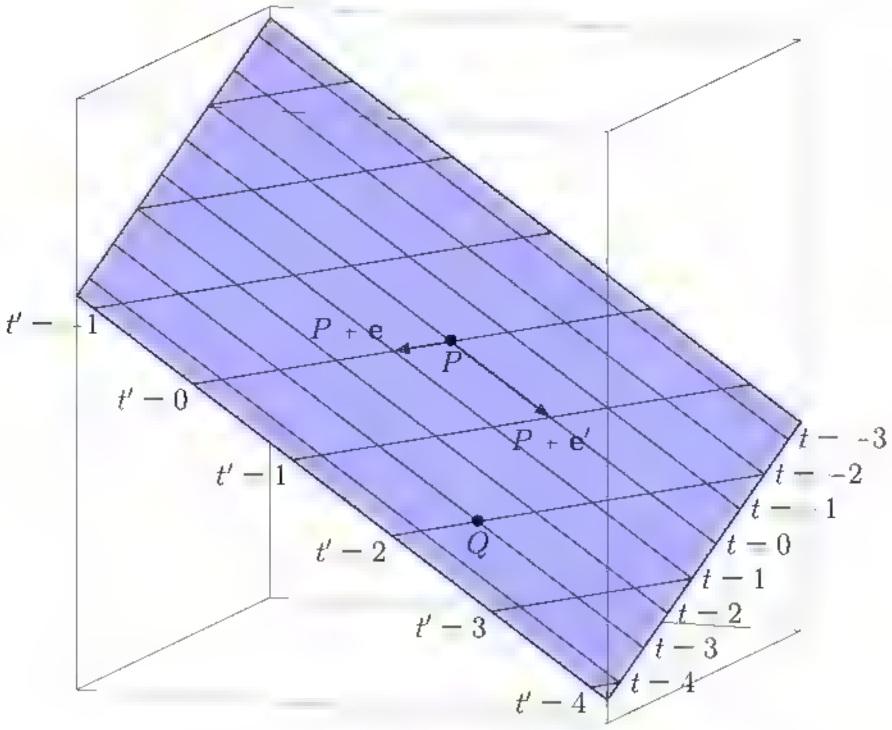


FIGURE 3.1.2. (b) A plane described by "parameters" t, t' via points $P + t\mathbf{e} + t'\mathbf{e}'$.

(b) (parametric form) The set of all vectors of the form $\begin{bmatrix} 1+2t \\ -t+3t' \\ 1-2t' \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + t\begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix} + t'\begin{bmatrix} 0 \\ 3 \\ -2 \end{bmatrix} =$ $P + t\mathbf{e} + t'\mathbf{e}'$, where t, t' take all possible scalar values. This is called a *parametric form*. The word "parametric" means that we have a formula which produces precisely the points on the plane and no others: substituting different values of t and t' – called "parameters" – gives different points on the plane, and every point in the plane is obtained from some value for t and t'. (Contrast this with (a), where we need to find a solution of this equation before we can obtain a point on the plane.) This yields a grid as in Figure 3.1.2 with lines corresponding to fixing the value of t or t'

(and $\mathbf{e}$ and $\mathbf{e}'$ are *displacement vectors*), for a reason we'll soon explain. The analogue for a line using one parameter t arose in Example 1.3.9 and will be discussed in general in Section 3.3.

Once we accept that both preceding methods describe the same plane, here are two more ways to characterize a plane that turn out in each case to yield the same plane as above.

(c) (point and normal vector form) The plane determined by the properties that it passes through the point $P = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ and is perpendicular to the direction $\mathbf{n} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ (this vector $\mathbf{n}$ is called a "normal vector" to the plane); see Figure 3.1.3.

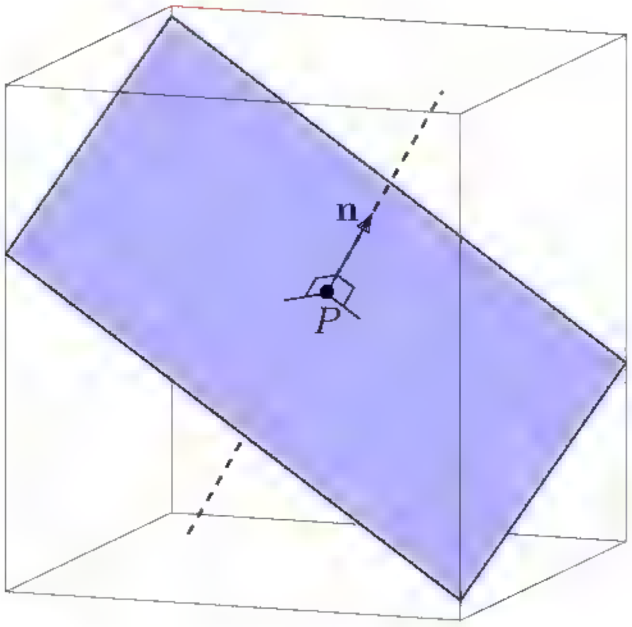


FIGURE 3.1.3. (c) The same plane, described using a point P and a "normal" (perpendicular) vector $\mathbf{n}$

(d) (three points on a plane) The plane determined by the fact that it passes through the three points $\begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}$, and $\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$. It is important that these three points do not all lie on the same common line in space; see Figure 3.1.4. (Later on we would like a *systematic* method – something that can be given as instructions to a computer – to show that these 3 points do not lie on a common line in space. We will address this in Section 3.2.)

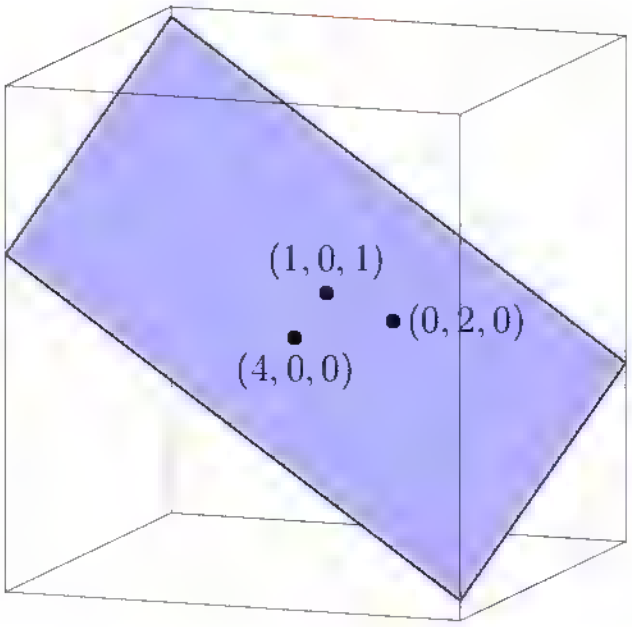


FIGURE 3.1.4. (d) The same plane, described as passing through 3 non-collinear points

These four descriptions are useful for different things:

- If we want to figure out the plane of the roof, we might measure the height at three corners of the house, and use description (d).
- To answer our Question at the start of this section – do points P and Q lie on the same side of the plane or not? – we can use description (a) (see Example 3.1.2).
- We will see in Section 3.2 that (c) amounts to a useful geometric interpretation of (a).
- The (x, y)-coordinates that describe points in coordinate plane geometry using the x and y coordinate axes amount to imposing a square grid and marking off distances from a pair of perpendicular reference lines (the x-axis and the y-axis). The parametric form in (b) corresponds to a way of imposing a (not necessarily square) grid on a plane which can be used in the same way to describe the locations of all points in the plane relative to the axes of this grid. The "parameters" t and t' play a role similar to that of x and y in coordinate plane geometry; e.g., in Figure 3.1.2 the point Q in the plane corresponds to $t = 3$ and $t' = 2$ due to the two grid lines that cross at Q, and in vector language this says $Q = P + 3\mathbf{e} + 2\mathbf{e}'$.

  Be careful however, since the analogy with (x, y)-coordinates in plane geometry is not quite perfect. As Figure 3.1.2 shows, the "t-axis" ($t' = 0$) emanating from P in the direction of the displacement $\mathbf{e}$ and the "t'-axis" ($t = 0$) emanating from P in the direction of the displacement $\mathbf{e}'$ might not be perpendicular to each other! The usefulness of this non-perpendicularity is related to visual experience in that if you look at a part of a plane from a very steep angle (rather than 'head-on') then your sense of geometry in that part of the plane may be distorted.

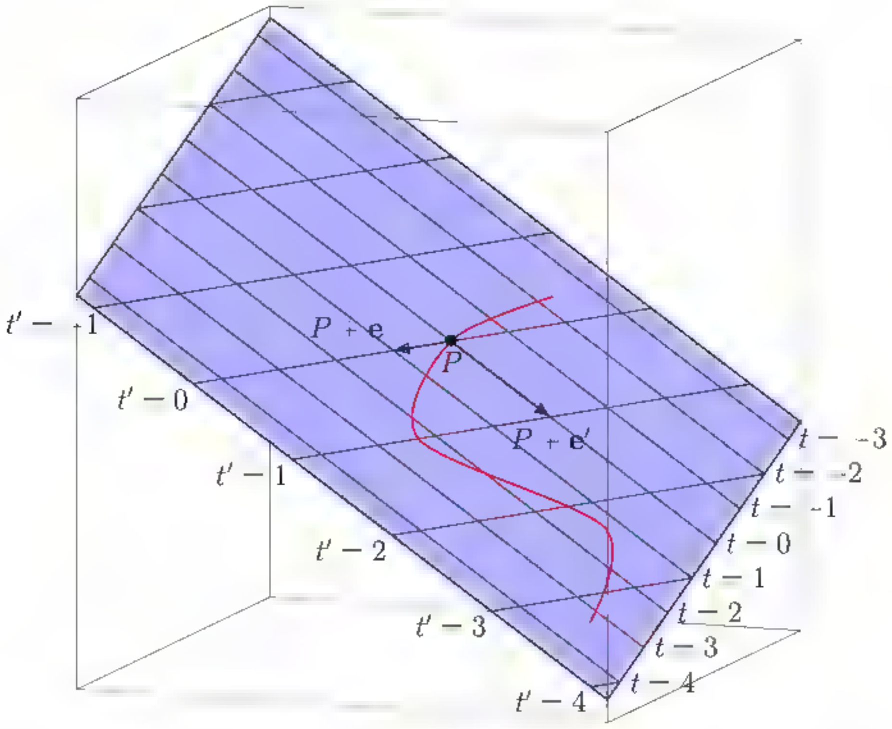


FIGURE 3.1.5. Varying the parameters t, t' continuously traces out a squiggle *inside* the plane

An advantage of the parametric form is that as we independently vary values of the parameters t and t', the vectors we get in the parametric form are guaranteed to lie *exactly* on the plane. For instance, if we want to trace out some path in the plane, then by varying the values of t and t' continuously we trace out a continuous curve exactly in the plane (see Figure 3.1.5).

Remark 3.1.1. The parametric description for planes and its analogues for more complicated surfaces (such as a sphere, a cylinder, etc.) is quite useful in computer graphics to generate the image of a path of motion lying exactly on a specific surface. For such applications a parametric form is *far more useful* than an equational form; as we have said before, with the parametric form we do not have to solve for anything.

Example 3.1.2. Coming back to the type of question at the start of this section with the roof of the house, how do we decide whether or not the points $(1, 1, 2)$ and $(-2, 1, 3)$ lie on the same side of a plane for which we have obtained an equation, such as $x + 2y + 3z = 4$?

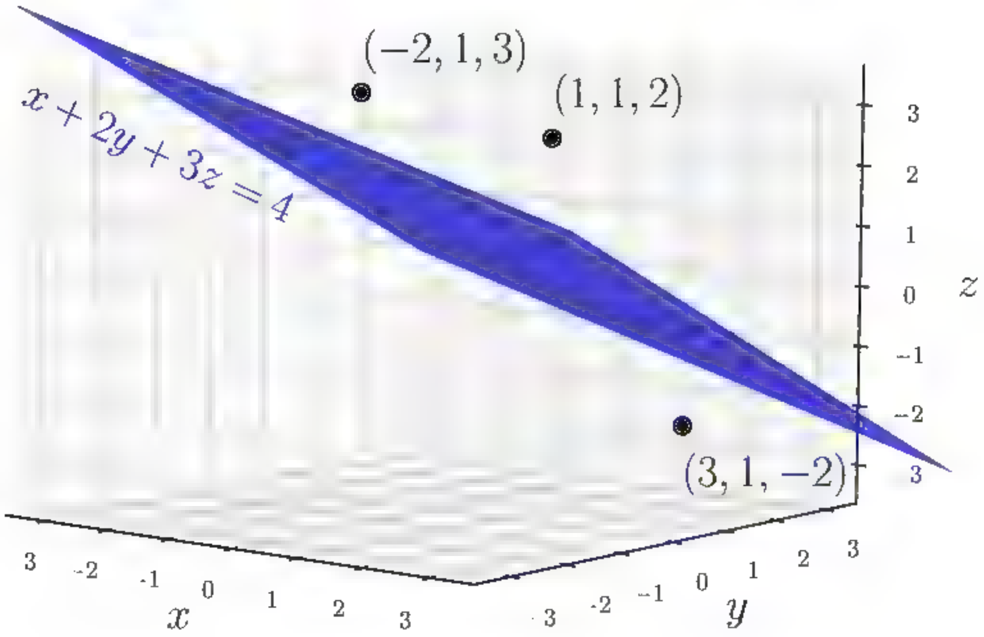


FIGURE 3.1.6. Points on either side of the same plane as in Figure 3.1.1, seen after applying an 80-degree horizontal rotation to the right and "zooming in" a bit.

One way to decide is: draw the solutions to the equation on a computer, as in Figure 3.1.6, and take a look! But the method of "taking a look" is not satisfactory because:

(i) in examples with much bigger numbers or many points at once, it could be impractical to have to create a picture and look at it;
(ii) we want a systematic method, without having to draw a picture in 3 dimensions (e.g., a computer program can't be told to "take a look").
iii) there are analogues of these questions involving "(hyper)planes" in $\mathbf{R}^n$ with huge $n > 3$, and their answers have applications in data science (e.g., the machine learning technique called "support vector machines"; see Example 19.4.4) and in economics (e.g., the separating hyperplane theorem, used to study profit functions in producer theory and to prove the second fundamental theorem in welfare economics). Developing intuition about the link between algebra of equations for planes and the geometry of how planes sit in $\mathbf{R}^3$ is very helpful for later work in $\mathbf{R}^n$ with $n > 3$.

Here is a better way. A plane divides $\mathbf{R}^3$ into two regions: the portion on one side of the plane and the portion on the other side. For the plane $x + 2y + 3z = 4$ as in Figure 3.1.6, one side is given by the inequality $x + 2y + 3z > 4$ (which is very closely related to the equation for the plane itself) and the other side of the plane is given by the opposite inequality $x + 2y + 3z < 4$. To see why this is reasonable, think about the analogue when we replace the equation $x + 2y + 3z = 4$ for Figure 3.1.6 with the much simpler equation $z = 0$ that describes the horizontal xy-plane in space. The region above this horizontal plane is where $z > 0$ and the region below it is where $z < 0$, as shown in Figure 3.1.7.

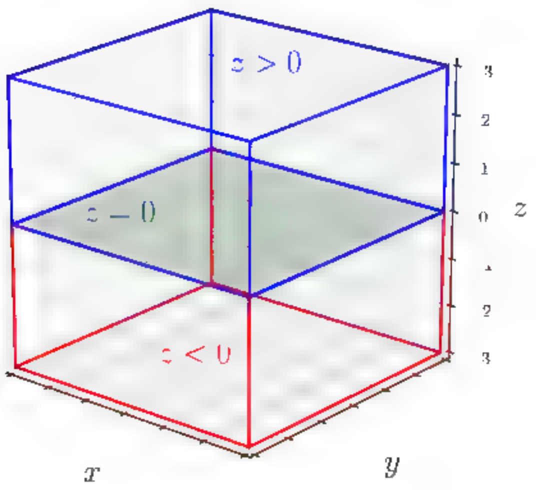


FIGURE 3.1.7. The regions $z > 0$ and $z < 0$ as opposite sides of the plane $z = 0$

Returning one last time to our original question, we can plug the two points $(1, 1, 2)$ and $(-2, 1, 3)$ into the expression $x + 2y + 3z$. Both satisfy the same direction of inequality: "$x + 2y + 3z > 4$", so these points lie on the same side of the plane. By contrast, the point $(3, 1, -2)$ satisfies $x + 2y + 3z < 4$, so it lies in the opposite side of the plane from those other two points $(1, 1, 2)$ and $(-2, 1, 3)$. These purely algebraic calculations match exactly what we see visually in Figure 3.1.6! ■

We next explain how to use vectors to transfer between the various preceding ways of describing a plane, by working through some examples in detail.

3.2. Worked examples. Suppose $\mathcal{P}$ is the plane in $\mathbf{R}^3$ going through the different points

$$(0, 1, 1), (0, 2, 3), (1, 3, 2)$$

that aren't all on a common line in $\mathbf{R}^3$ (as we will explain how to verify in a systematic manner in a moment). We seek: (i) a parametric form for $\mathcal{P}$, (ii) a normal vector to $\mathcal{P}$, and (iii) an equation for $\mathcal{P}$.

To get started, let's name each of the points, using vector notation:

$$P = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, Q = \begin{bmatrix} 0 \\ 2 \\ 3 \end{bmatrix}, R = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix}. \tag{3.2.1}$$

Thus, the displacement vectors from P to Q, and from P to R (i.e., difference vectors), are:

$$\overrightarrow{PQ} = Q - P = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}, \quad \overrightarrow{PR} = R - P = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}. \tag{3.2.2}$$

Remark 3.2.1. We claim that the only way three different points can be on a common line in space is when the difference vectors from one of them to the other two lie along the same or opposite directions. This corresponds to those two difference vectors being scalar multiples of each other (a positive scalar when pointing in the same direction, and a negative scalar when pointing in opposite directions). To explain this visually, in Figure 3.2.1 we have drawn two possible pictures of three collinear points P', Q', R'. In the first picture $\overrightarrow{P'Q'}$ and $\overrightarrow{P'R'}$ are positive multiples of each other, while in the second picture $\overrightarrow{P'Q'}$ and $\overrightarrow{P'R'}$ are negative multiples of each other.

Clearly the difference vectors in (3.2.2) aren't scalar multiples of each other. Thus, by Remark 3.2.1, indeed P, Q, R in (3.2.1) are not on a common line. This conclusion can also be verified by making a careful 3-dimensional picture, but the purpose of the algebraic method of difference vectors as just implemented is that it is more systematic and much easier to carry out in practice (and it is the method that has to be used by a computer too).

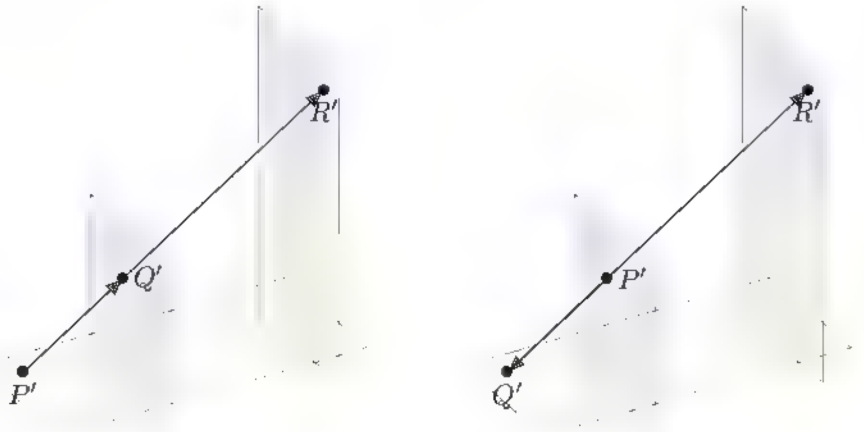


FIGURE 3.2.1. Two configurations of three collinear points P', Q', and R'

(i) **(parametric form)** By thinking visually, we can get to every point on the plane $\mathcal{P}$ by:

- starting at the point P,
- walking, for some specific distance, in the direction of $\overrightarrow{PQ}$,
- then walking, for some specific distance, in the direction of $\overrightarrow{PR}$.

In mathematical symbols, this says that the plane consists of all vectors of the form

$$P + t\overrightarrow{PQ} + t'\overrightarrow{PR} \tag{3.2.3}$$

for scalars t, t' (with each vector in the plane obtained for a uniquely determined pair of values (t, t')). Plugging in (3.2.1) and (3.2.2), $\mathcal{P}$ consists of all points of the form

$$\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + t \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} + t' \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ t \\ 2t \end{bmatrix} + \begin{bmatrix} t' \\ 2t' \\ t' \end{bmatrix} = \begin{bmatrix} t' \\ t + 2t' + 1 \\ 2t + t' + 1 \end{bmatrix}. \tag{3.2.4}$$

This is a parametric form of the plane $\mathcal{P}$: if we plug in various values for t, t' (the "parameters") then we get various points on the plane, and we obtain *all* points of the plane in this way. Visually, this parametric form corresponding to laying out a "grid" on the plane $\mathcal{P}$ as in Figure 3.2.2 with "origin" at P, t-axis pointing along $\mathbf{e} = \overrightarrow{PQ}$, and t'-axis pointing along $\mathbf{e}' = \overrightarrow{PR}$.

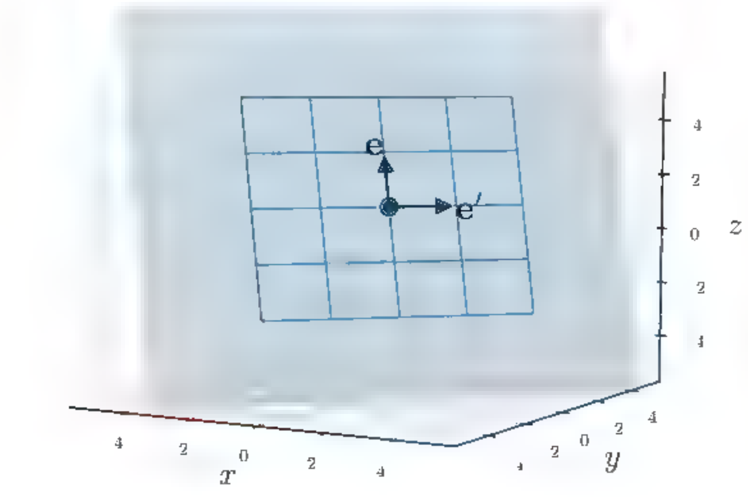


FIGURE 3.2.2. A grid arising from a parametric form, with displacements in directions e and e′

(ii) (**normal vector**) Next, let us find a *normal vector*: a nonzero vector $\mathbf{n} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ that is perpendicular to all directions along the plane (i.e., differences between points in the plane). Since vectors are perpendicular exactly when their dot product is zero, we want $\mathbf{n} \cdot \overrightarrow{PQ} = 0$ and $\mathbf{n} \cdot \overrightarrow{PR} = 0$; equivalently, $\mathbf{n} \cdot \mathbf{e} = 0, \mathbf{n} \cdot \mathbf{e}' = 0$. (We do **not** solve this via the "cross product" in $\mathbf{R}^3$ because a fundamental principle of this book is that we only use techniques that will adapt to $\mathbf{R}^n$ for *every* n; see Remark 1.5.1.) Via the numbers in (3.2.2), our task says

$$\mathbf{n} \cdot \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} = 0, \quad \mathbf{n} \cdot \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = 0, \tag{3.2.5}$$

which explicitly means

$$b + 2c = 0, \quad a + 2b + c = 0. \tag{3.2.6}$$

We will study how to solve such systems of equations *systematically* in any number of variables later on (in Chapter 21). But this situation is "small" enough in complexity (2 equations with 3 unknowns) that we can do it by hand as follows. From the first equation in (3.2.6), we solve for b in terms of c to get $b = -2c$. Putting this into the second equation in (3.2.6), we get

$$a - 4c + c = 0.$$

Therefore, $a = 3c$ and $b = -2c$. It follows that vectors of the form $\mathbf{n} = \begin{bmatrix} 3c \\ -2c \\ c \end{bmatrix}$ are precisely those perpendicular to our plane (and $\mathbf{n} \neq \mathbf{0}$ when $c \neq 0$). Since any nonzero value of c will do, we may as well take $c = 1$. So $\begin{bmatrix} 3 \\ -2 \\ 1 \end{bmatrix}$ is a normal vector to the plane (as is any nonzero scalar multiple of it); see Figure 3.2.3.

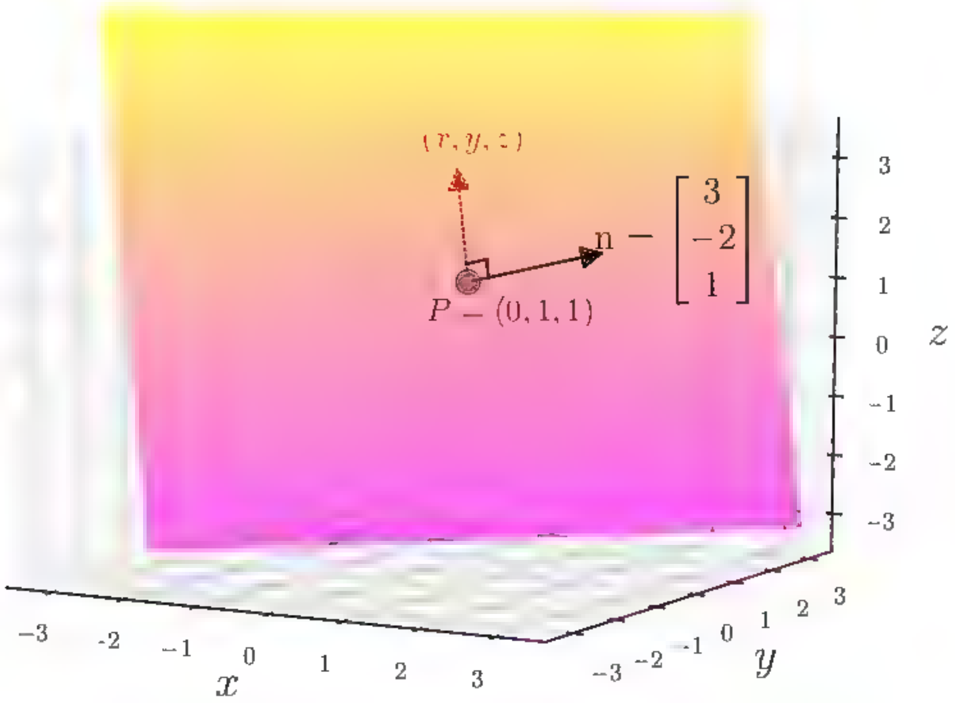


FIGURE 3.2.3. A normal vector to the plane

Instead of solving for b and a in terms of c as we did above (by plugging one equation into the other), we could have proceeded differently, such as solving in terms of a. For instead, the second equation gives $c = -2b - a$ in terms of a and b, and plugging this into c in the first equation gives

$$0 = b + 2c = b + 2(-2b - a) = b - 4b - 2a = -3b - 2a,$$

so $b = -(2/3)a$. Hence, $c = -2b - a = -2(-(2/3)a) - a = (4/3)a - a = a/3$, so we arrive at a description of normal vectors as

$$\mathbf{n}' = \begin{bmatrix} a \\ -(2/3)a \\ a/3 \end{bmatrix} = a \begin{bmatrix} 1 \\ -2/3 \\ 1/3 \end{bmatrix}.$$

This is essentially the same as the description of possible $\mathbf{n}$'s above except that we have multiplied $\begin{bmatrix} 3 \\ -2 \\ 1 \end{bmatrix}$ by $1/3$.

(iii) (**equational form**) Finally, we want to find an equation for the plane. We can do this with the normal vector: if $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$ is any point on the plane, then the vector from $P = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ to $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$ (i.e., the *difference* between these two vectors) must be perpendicular to $\mathbf{n}$, simply because $\mathbf{n}$ is a normal vector to the plane. That means

$$\begin{bmatrix} x - 0 \\ y - 1 \\ z - 1 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ -2 \\ 1 \end{bmatrix} = 0,$$

or equivalently

$$3x - 2(y - 1) + z - 1 = 0.$$

Hence, cleaning up the algebra a bit, an equation for the plane is

$$3x - 2y + z = -1.$$

To summarize:

(i) a parametric form of the plane is given by the vectors $\begin{bmatrix} t' \\ t + 2t' + 1 \\ 2t + t' + 1 \end{bmatrix}$ for varying scalars t, t',

(ii) a normal vector to the plane is $\begin{bmatrix} 3 \\ -2 \\ 1 \end{bmatrix}$ (with the plane passing through the point $P = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$),

(iii) an equation of the plane is $3x - 2y + z = -1$.

Keep in mind that we used the normal vector to find the equation, and we used dot products (against difference vectors) to detect perpendicularity. These two principles are applicable for passing among different descriptions of any plane in $\mathbf{R}^3$.

Example 3.2.2. Now let's hone our new skills for passing between algebra and geometry by working out another example, this time using only our brains and no pictures. Let's describe the equation of a plane in $\mathbf{R}^3$ that goes through the origin and is perpendicular to $\begin{bmatrix} 2 \\ 1 \\ 8 \end{bmatrix}$. (If you feel that it is too special to consider a plane that goes through the origin, another example of this flavor not passing through the origin is worked out in Example 3.2.5.)

If (x, y, z) is an arbitrary point on the plane, then the line segment connecting the origin to (x, y, z) must be perpendicular to $\begin{bmatrix} 2 \\ 1 \\ 8 \end{bmatrix}$. This says exactly that

$$\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) \cdot \begin{bmatrix} 2 \\ 1 \\ 8 \end{bmatrix} = 0.$$

In other words,

$$2x + y + 8z = 0. \tag{3.2.7}$$

This is the equation of the plane, and we read off from the coefficients that $(2, 1, 8)$ is a normal vector (unsurprising, given how the plane was originally defined). ■

Example 3.2.3. Next, we find a parametric form for the plane studied in Example 3.2.2, taking as our input the equation (3.2.7) that we have determined for this plane. We apply the method used earlier in this section to arrive at the parametric form (3.2.4) for the plane passing through the 3 points in (3.2.1). That is, we compute an expression looking like (3.2.3) via difference vectors based on a choice of 3 different points P, Q, R in the plane that aren't on a common line.

How do we choose 3 such points? One way is to simply pick some solutions to the equation (3.2.7), as we'll soon do in a systematic way, and make sure we make choices "randomly" enough that we avoid the 3 points all lying on the same line (as will essentially never happen unless we make unusually special choices, and we will make sure we have avoided that outcome by considering difference vectors as before).

In the present case there is an especially convenient solution to (3.2.7): the origin

$$P = (0, 0, 0)$$

in $\mathbf{R}^3$. This works because the constant on the right side of (3.2.7) is 0. What do we do to pick Q and R? An especially convenient method (which works rather generally) is to simply set one of the variables in the plane's equation to be 0, another to be 1, and to solve for the remaining variable.

For instance, if we set $z = 0, y = 1$ then solving for x in (3.2.7) gives the condition

$$0 = 2x + 1 + 8(0) = 2x + 1,$$

or $x = -1/2$. That is, we get the point

$$Q = (-1/2, 1, 0).$$

Similarly, if we set $y = 0, x = 1$ then solving for z in (3.2.7) gives the condition $2 + 8z = 0$, or $z = -1/4$, yielding the point

$$R = (1, 0, -1/4).$$

Since P is the origin, the difference vectors $\overrightarrow{PQ}$ and $\overrightarrow{PR}$ are nothing other than Q and R by another (vector) name, and by inspection these differences aren't scalar multiples of each other, so these 3 points P, Q, R don't lie on a common line! Hence, these 3 points will be appropriate for making a parametric form.

Remark 3.2.4. Before we work out the parametric form from these difference vectors, we briefly digress to address the choices we made to make the points Q and R above. We could have instead used $x = 0, z = 1$ to get $y = -8$, yielding yet another point $S = (0, -8, 1)$ which works just as well in place of Q or R. Similarly, we could have used $x = 0, y = 1$ to get $z = -1/8$ and hence the point $T = (0, 1, -1/8)$ that can be used just as well. There are zillions of ways to pick a point on the plane once we have an equation, by picking two of its coordinates at random and solving for the third using the equation.

> The method of finding points in the plane by setting one of x, y, z to be 0, another to be 1 (or really setting two of the variables to be whatever different random values we like), and solving for the third variable (using an equation of the plane) requires some care: we should make sure that the coefficient of that third variable solved at the end is not 0 (so the required division can be done). As long as the equation of the plane involves all 3 variables (i.e., we avoid planes in $\mathbf{R}^3$ such as $2x - 5z = 0$ in which some variable doesn't appear), then this issue never arises.

Coming back to our task with the points P, Q, R, since $P = \mathbf{0}$ the parametric form as in (3.2.3) is

$$P + t\overrightarrow{PQ} + t'\overrightarrow{PR} = tQ + t'R = t\begin{bmatrix}-1/2\\1\\0\end{bmatrix} + t'\begin{bmatrix}1\\0\\-1/4\end{bmatrix} = \begin{bmatrix}-t/2\\t\\0\end{bmatrix} + \begin{bmatrix}t'\\0\\-t'/4\end{bmatrix} = \begin{bmatrix}-t/2+t'\\t\\-t'/4\end{bmatrix}.$$

Concretely, as we vary the values of t and t' we sweep out precisely the points in the plane of interest.

If we made another choice of the 3 points in the plane, the resulting parametric form would look very different! For example, if we replace R with the point S found in Remark 3.2.4, the triple P, Q, S (again with $P = \mathbf{0}$) works just as well (i.e., P, Q, S are not all on a common line) and we obtain a parametric form in vector notation

$$tQ + t'S = t\begin{bmatrix}-1/2\\1\\0\end{bmatrix} + t'\begin{bmatrix}0\\-8\\1\end{bmatrix} = \begin{bmatrix}-t/2\\t-8t'\\t'\end{bmatrix}.$$

■

Example 3.2.5. Let's find the equation of the plane in $\mathbf{R}^3$ that goes through $P = (1, 2, 3)$ and is perpendicular to $\mathbf{n} = \begin{bmatrix}3\\4\\5\end{bmatrix}$. A picture of this is given in Figure 3.2.4.

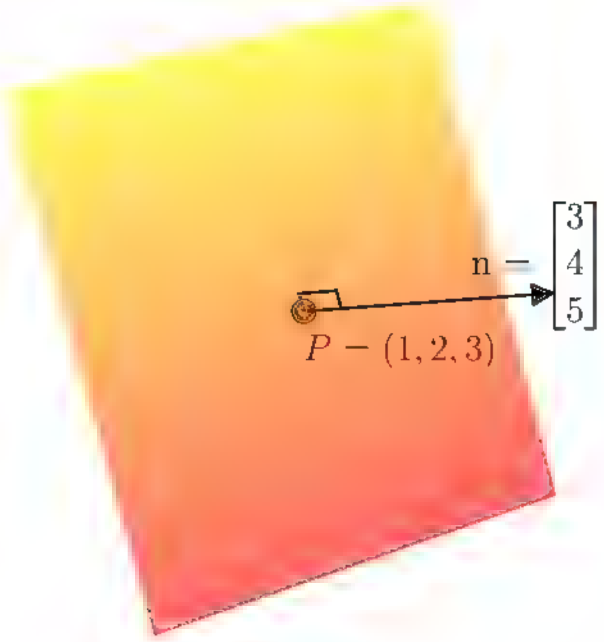


FIGURE 3.2.4. A plane through a given point, with a given normal vector

The method by which we will find the equation is based on geometric reasoning (in terms of perpendicularity to difference vectors), but we emphasize that the method *does not require us to draw this picture* of the plane, nor does having the exact drawing (as we do) play any role in figuring out the equation.

If (x, y, z) is an arbitrary point on the plane, then the line segment connecting $(1, 2, 3)$ to (x, y, z) must be perpendicular to $\begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix}$. This says exactly that

$$\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}\right) \cdot \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} = 0.$$

In other words,

$$3(x-1) + 4(y-2) + 5(z-3) = 0.$$

This is the equation of the plane. This equation can be simplified by multiplying out to get

$$3x - 3 + 4y - 8 + 5z - 15 = 0,$$

and then combining the constants to obtain the cleaner form

$$3x + 4y + 5z = 26. \tag{3.2.8}$$

From the coefficients we again read off that $(3, 4, 5)$ is a normal vector to the plane (*not* to the individual points in the plane, but rather to *differences* between such points). This is no surprise, in view of how the plane was originally defined! ■

Example 3.2.6. Let's now get a parametric equation of the plane in Example 3.2.5 whose equation we found in (3.2.8). We know the plane goes through the vector $P = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$. Following the procedure we have gone through a couple of times already based on (3.2.3), we need to find two additional vectors Q and R in the plane different from P so that these 3 points in the plane aren't on a common line, or equivalently so that the difference vectors $Q - P$ and $R - P$ are not scalar multiples of each other.

In other words, we seek different solutions Q and R to the equation of the plane in (3.2.8) so that $Q - P$ and $R - P$ are not scalar multiples of each other. Once we find such Q and R, a parametric form of the plane is given by vectors of the form

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + t\left(Q - \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}\right) + t'\left(R - \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}\right)$$

where t, t' are any real numbers.

From the equation $3x + 4y + 5z = 26$ let's make candidates for Q and R by setting one variable to be 0, another to be 1, and solving for the third variable:

Setting $y = 0$ and $z = 1$ gives $3x + 5 = 26$, or $x = 7$, yielding $Q = (7, 0, 1)$.

Setting $z = 0$ and $x = 1$ gives $3 + 4y = 26$, or $y = 23/4$, yielding $R = (1, 23/4, 0)$.

This Q and R "work" in the sense that the difference vectors

$$Q - P = \begin{bmatrix} 7 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 6 \\ -2 \\ -2 \end{bmatrix}, \quad R - P = \begin{bmatrix} 1 \\ 23/4 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 0 \\ 15/4 \\ -3 \end{bmatrix}$$

are not scalar multiples of each other (why?). Thus, a parametric form of the plane is

$$P + t\overrightarrow{PQ} + t'\overrightarrow{PR} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + t \begin{bmatrix} 6 \\ -2 \\ -2 \end{bmatrix} + t' \begin{bmatrix} 0 \\ 15/4 \\ -3 \end{bmatrix} = \begin{bmatrix} 1+6t \\ 2-2t+15t'/4 \\ 3-2t-3t' \end{bmatrix}$$

for any real numbers t, t'. ■

3.3. Parametric form and dimension. In all of the preceding examples for which we have found a parametric form for a plane, there are always 2 parameters t and t'. This is an algebraic incarnation of the familiar geometric idea that a plane is 2-dimensional: the two values t and t' keep track of a point's position in the plane relative to a pair of reference lines through a common point P (i.e., the lines through $\overrightarrow{PQ}$ and $\overrightarrow{PR}$ that meet at P). In Chapter 4 we take up this idea of "dimension" in a wider context, where it will become a powerful tool for applying our "visual intuition" to real-world situations with $\mathbf{R}^n$ for very large n (far beyond the $\mathbf{R}^3$ of daily experience).

Since we have discussed how to describe a plane in $\mathbf{R}^3$ (such as a plane through 3 given points) in terms of 2 parameters, let's drop the dimension by 1 and explain how to describe a line in $\mathbf{R}^3$ using 1 parameter; this will work equally well for lines in $\mathbf{R}^2$, putting Example 1.3.9 into a wider context. There are two ways to think about a line in $\mathbf{R}^3$:

(i) passing through a specified point $\mathbf{p}$ in the direction of a nonzero vector $\mathbf{v}$ (see Figure 3.3.1),
(ii) passing through 2 given different points $\mathbf{p}$ and $\mathbf{q}$ (see Figure 3.3.2).

We shall see how to describe the line in both situations.

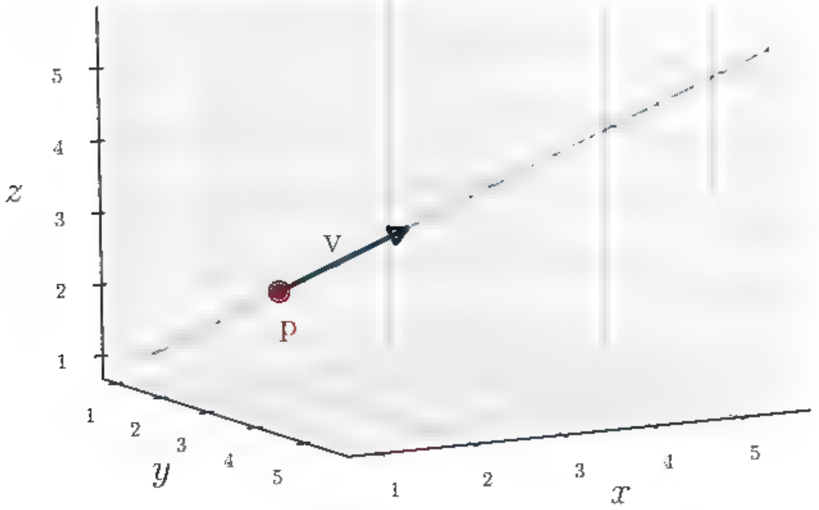


FIGURE 3.3.1. Method (i): line in space through a given point and with a given direction

First consider a line ℓ passing through $\mathbf{p}$ in the direction of a vector $\mathbf{v}$, as in Figure 3.3.1. This means that if we begin at $\mathbf{p}$ and move along the line ℓ to a point $\mathbf{x}$ on the line then we should have moved "in the direction of $\mathbf{v}$" (or $-\mathbf{v}$). In algebraic terms, this says that the difference vector $\mathbf{x} - \mathbf{p}$ should be a scalar multiple of $\mathbf{v}$. That is: $\mathbf{x} - \mathbf{p} = t\mathbf{v}$ for some scalar t, or equivalently

$$\mathbf{x} = \mathbf{p} + t\mathbf{v}. \tag{3.3.1}$$

As we vary t, we sweep out the entirety of the line (with $t > 0$ corresponding to motion along ℓ away from $\mathbf{p}$ in the direction of $\mathbf{v}$, and $t < 0$ corresponding to motion along ℓ away from $\mathbf{p}$ in the direction of $-\mathbf{v}$).

Example 3.3.1. As a numerical example, if $\mathbf{p} = \begin{bmatrix} 1 \\ -5 \\ 2 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} -3 \\ 2 \\ 7 \end{bmatrix}$ then the line ℓ through $\mathbf{p}$ in the direction of $\mathbf{v}$ consists of the points

$$\mathbf{p} + t\mathbf{v} = \begin{bmatrix} 1 \\ -5 \\ 2 \end{bmatrix} + t \begin{bmatrix} -3 \\ 2 \\ 7 \end{bmatrix} = \begin{bmatrix} 1 \\ -5 \\ 2 \end{bmatrix} + \begin{bmatrix} -3t \\ 2t \\ 7t \end{bmatrix} = \begin{bmatrix} 1-3t \\ -5+2t \\ 2+7t \end{bmatrix}$$

for varying scalars t. ■

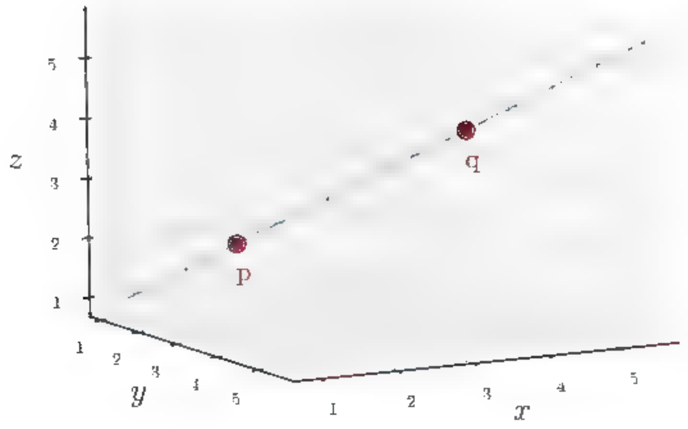


FIGURE 3.3.2. Method (ii): line in space through 2 different points

Next, suppose ℓ is described as the line passing through different points $\mathbf{p}$ and $\mathbf{q}$ as in Figure 3.3.2. The (nonzero) difference vector $\mathbf{v} = \mathbf{q} - \mathbf{p}$ points along the direction of the line, as shown in Figure 3.3.3. It follows that ℓ is the line through $\mathbf{p}$ in the direction of $\mathbf{v}$. Hence, (3.3.1) gives us a parametric form:

$$\mathbf{x} = \mathbf{p} + t(\mathbf{q} - \mathbf{p}) = (1-t)\mathbf{p} + t\mathbf{q} \tag{3.3.2}$$

with varying scalars t. For $t = 0$ this recovers $\mathbf{p}$ on the line, for $t = 1$ it recovers $\mathbf{q}$ on the line, and for $0 \leq t \leq 1$ we get the points of the segment joining $\mathbf{p}$ and $\mathbf{q}$, with $t = 1/2$ corresponding to the midpoint.

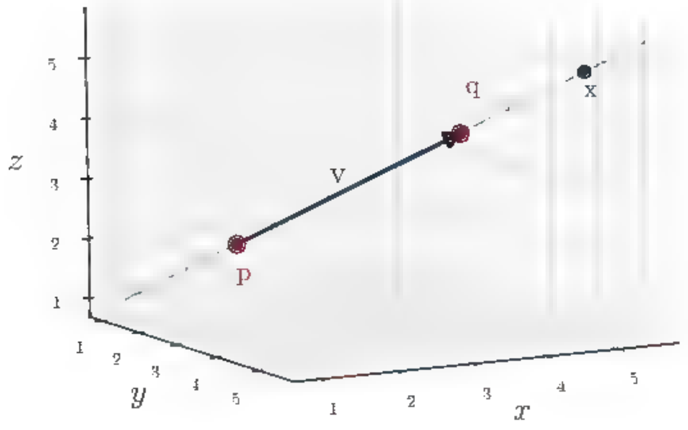


FIGURE 3.3.3. Difference vector along a line

Example 3.3.2. As a numerical example, if $\mathbf{p} = \begin{bmatrix} 3 \\ 1 \\ -1 \end{bmatrix}$ and $\mathbf{q} = \begin{bmatrix} 2 \\ 4 \\ 5 \end{bmatrix}$ then the line through $\mathbf{p}$ and $\mathbf{q}$ consists of the points

$$(1-t)\mathbf{p} + t\mathbf{q} = \begin{bmatrix} 3(1-t) \\ 1-t \\ t-1 \end{bmatrix} + \begin{bmatrix} 2t \\ 4t \\ 5t \end{bmatrix} = \begin{bmatrix} 3-t \\ 1+3t \\ -1+6t \end{bmatrix}$$

for varying scalars t. ■

3.4. Equation for a plane as a collection of equations for lines. In this optional section we use a "slicing" technique to explain why equations of the form $ax + by + cz = d$ (with at least one of $a, b, c \neq 0$) are planes, building on our experience with equations of lines in $\mathbf{R}^2$.

Let us first rearrange the variables if necessary so that $c \neq 0$. Dividing through by c doesn't affect the solution set to the equation in $\mathbf{R}^3$ (just as the equations $2x + 3y = 7$ and $4x + 6y = 14$ and $(2/3)x + y = 7/3$ all describe the same line in $\mathbf{R}^2$), so dividing by c, we have brought the equation to a form where the coefficient of the variable z equals 1. For example, if we began with $2x - 3y + 4z = 8$ then we would arrive at the equation $(1/2)x - (3/4)y + z = 2$.

Now bring the terms involving x and y to the other side. Our task is now to explain why equations of the form

$$z = a'x + b'y + c'$$

(with some new constants a', b', c') define planes in $\mathbf{R}^3$. Let us consider a specific example.

Example 3.4.1. We visualize what is going on by considering

$$z = -2x - y + 3. \tag{3.4.1}$$

This is the blue plane on the left in Figure 3.4.1, and we would like to see how our knowledge about 2-variable equations that describe lines explains *why* this 3-variable equation really does describe a plane. We want to relate the algebra of the equation (3.4.1) to the geometry of the solution set in $\mathbf{R}^3$ (which we'll suppose we haven't already drawn on a computer to make Figure 3.4.1, since we want to understand *why* equations such as (3.4.1) always define a plane). The key idea is to consider what happens for each *specific* value for x, slicing the solution set by vertical planes $x = C$.

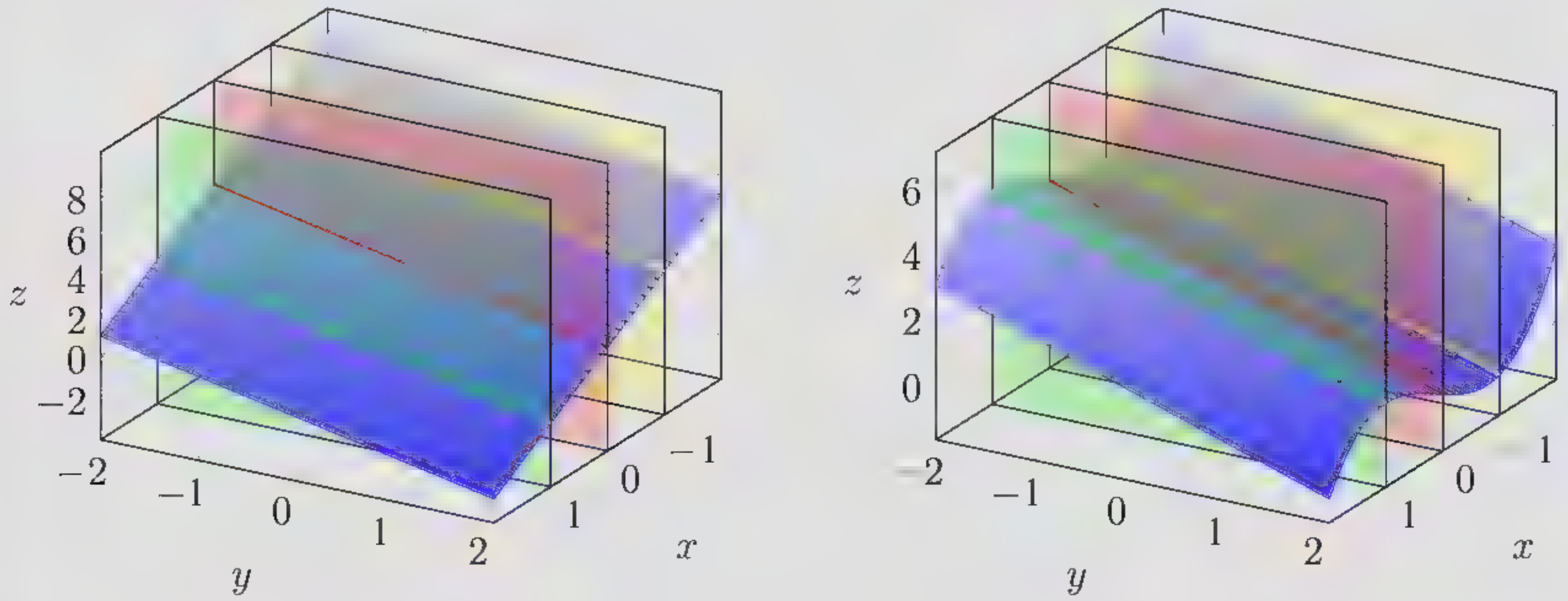


FIGURE 3.4.1. A plane and wavy surface each assembled from lines of slope -1

Consider where the solution set to (3.4.1) intersects the (red) vertical plane $x = 0$. We can find this intersection algebraically by setting x to equal 0 in the initial equation to arrive at the equation

$$z = -y + 3;$$

we recognize this as a line of slope -1 in the plane $x = 0$; this is precisely the set where the red vertical plane $x = 0$ on the left in Figure 3.4.1 slices the blue solution set to (3.4.1). Likewise, consider where the green vertical plane $x = 1$ slices the blue solution set to (3.4.1). As before, we can find this algebraically by setting x to be 1 in the equation (3.4.1) to arrive at

$$z = -y + 1$$

in the plane $x = 1$. We recognize this as another line, still of slope -1, but shifted 2 *units down* from the previous line (since $1 = 3 - 2$). In general, for any *specific* number h, the vertical plane $x = h$ cuts the blue solution set in the part of that plane given by the equation

$$z = -y + (-2h + 3),$$

and we recognize this as a line of slope -1 in the plane $x = h$. *Assembling these lines together as h varies* reconstructs the entire solution set to (3.4.1).

We now face the question of what this assembly of lines with a common slope looks like. Does one always get a *plane* (as on the left side in Figure 3.4.1) by assembling lines of common slope in the collection of parallel vertical planes $x = h$ as h varies? Actually, the answer is no! Perhaps surprisingly, one can "disassemble" a wavy surface into a collection of slices which are lines all of the same slope, as on the right in Figure 3.4.1. So we require some further property beyond just that the slices are lines with the same slope.

Knowing that a region in $\mathbf{R}^3$ meets each plane $x = h$ in a line with common slope (as for both options in Figure 3.4.1) says that it is given by an equation of the form $cz = by + f(h)$ for some scalars b, c, where $f(h)$ is the value of the "z-intercept"; i.e., where this line intersects the z-axis in the plane $x = h$ with its coordinates (y, z). The fact that these lines fit together to make a *flat* surface corresponds to the fact that the "heights" of these lines change at a "uniform rate" as we vary the value h. This "uniform rate" for the heights says that $f'(h)$ is constant, which says that $f(h) = ah + d$ for some constants a and d. Hopefully it is not too confusing now to remember that h is just any one of the values that x can take, so let us just write x instead of h to get $cz = by + (ax + d)$, or equivalently $ax + by + (-c)z = -d$. The wavy surface on the right in Figure 3.4.1 is what happens when $f(x)$ is the more complicated function $2 - (1/2)x^3 + (1/6)x^2 + (7/6)x$ and $z = -y + f(x)$. ■

Chapter 3 highlights (links to highlights in [illegible] and next chapters)

Notation	Meaning	Location in text
$P+t\mathbf{v}+t'\mathbf{v}'$ for varying t, t'	parametric description of plane in $\mathbf{R}^3$ through point P in directions of $\mathbf{v}$ and $\mathbf{v}'$	(3.2.3), (3.2.4)
$\mathbf{p}+t\mathbf{v}$ for varying t	parametric description of line in $\mathbf{R}^3$ (or $\mathbf{R}^2$) through point $\mathbf{p}$ in direction of $\mathbf{v}$	(3.3.1)
$t\mathbf{p}+(1-t)\mathbf{q}$ for varying t	parametric description of line through distinct points $\mathbf{p}$ and $\mathbf{q}$ (varying scalar t)	(3.3.2)

Concept	Meaning	Location in text
equational form of a plane in $\mathbf{R}^3$	equation $ax+by+cz=d$ (with at least one of a, b, c nonzero) whose solutions are the points of the plane	Section 3.1(a), Figure 3.1.1
parametric form of a plane in $\mathbf{R}^3$	describes points in terms of moving away from a single point in plane, along two “independent” directions in the plane	Section 3.1(b), Figure 3.1.2
point-normal form of a plane in $\mathbf{R}^3$	description in terms of point in plane and (nonzero) normal vector to plane	Section 3.1(c), Figure 3.1.3
normal vector to a plane in $\mathbf{R}^3$	nonzero vector orthogonal to all *differences* between points in the plane	Section 3.2(ii)

Result	Meaning	Location in text
relate normal vector to equation for plane in $\mathbf{R}^3$	for equation $ax+by+cz=d$, normal vector is $\begin{bmatrix} a \\ b \\ c \end{bmatrix}$	Examples 3.2.2, 3.2.5
parametric forms of a line in $\mathbf{R}^3$ (or $\mathbf{R}^2$)	geometric interpretation and algebraic description of two ways to describe a line in terms of some vector data and a varying scalar t	Figure 3.3.1 and (3.3.1), as well as Figure 3.3.2 and (3.3.2)
relate number of parameters and dimension	parametric forms of line in $\mathbf{R}^3$ use 1 parameter, parametric form of plane in $\mathbf{R}^3$ uses 2 parameters	Section 3.3

Skill	Location in text
transition among the 4 different ways of describing a plane in $\mathbf{R}^3$ (including “three non-collinear points”)	Section 3.2
find nonzero 3-vector perpendicular to two given nonzero 3-vectors by using 2 equations in 3 unknowns (right viewpoint for later in $\mathbf{R}^n$)	Section 3.2(ii) (see (3.2.5), (3.2.6))
using equational form of plane in $\mathbf{R}^3$, determine if 2 points not in plane are on same side or opposite sides	Example 3.1.2
interpret parametric form of plane in $\mathbf{R}^3$ in terms of grid (not expected to produce the picture, but know typical visualizations as in Figures 3.1.2, 3.2.2)	Section 3.2(i)
describe line in $\mathbf{R}^3$ (or $\mathbf{R}^2$) in parametric form given either a point on the line and a nonzero direction (or displacement) along the line, or two distinct points on the line	Examples 3.3.1, 3.3.2

4. Span, subspaces, and dimension

A theme of this book is taking geometric concepts (such as length or angle) and generalizing them *usefully* to n-vectors for any n. In this chapter, we describe how to generalize the geometric concept of plane in $\mathbf{R}^3$ to appropriate subsets of $\mathbf{R}^n$ for any n. We will see an application in Chapter 7 where we use geometry in $\mathbf{R}^n$ to study the "least squares" method to fit a line to n data points in $\mathbf{R}^2$, the first of many applications we will encounter for geometric intuition in the context of multivariable problems.

The theme of this chapter is *linear subspaces* of $\mathbf{R}^n$. By the end of the chapter, you should be able to:

- define span, linear subspace, and dimension in $\mathbf{R}^n$;
- recognize linear subspaces of $\mathbf{R}^3$ and their dimensions;
- relate orthogonality to 1 or 2 given vectors in $\mathbf{R}^n$ to a linear subspace.

The *geometric intuition* acquired via the concept of linear subspace is useful in engineering, economics, computer science, data science, physics, etc., as we will see throughout the rest of the book.

4.1. Span and linear subspaces. Consider a plane $\mathcal{P}$ in $\mathbf{R}^3$ passing through $\mathbf{0} = (0, 0, 0)$. We want to express mathematically the idea that $\mathcal{P}$ is "flat" with "two degrees of freedom". Choose two other points in $\mathcal{P}$, denoted $\mathbf{v}$ and $\mathbf{w}$, that *do not lie on a common line through* $\mathbf{0}$. (This requirement is akin to the condition that the nonzero difference vectors $\overrightarrow{PQ}$ and $\overrightarrow{PR}$ in Section 3.2 are not scalar multiples of each other.)

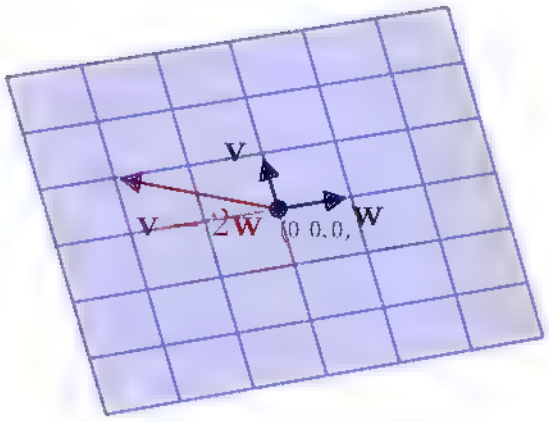


FIGURE 4.1.1. Reaching anywhere in a plane $\mathcal{P}$ by combining steps in two directions

As we saw in Section 3.2, we can get to any point on $\mathcal{P}$ by starting at $\mathbf{0}$ and walking first some specific distance in the $\mathbf{v}$-direction, then some specific distance in the $\mathbf{w}$-direction. For instance, Figure 4.1.1 shows the overall effect of walking a distance of $\|\mathbf{v}\|$ in the direction of $\mathbf{v}$ and a distance of $2\|\mathbf{w}\|$ in the direction of $-\mathbf{w}$. Symbolically,

$$\mathcal{P} = \{\text{all vectors of the form } a\mathbf{v} + b\mathbf{w}, \text{ for scalars } a, b\} \tag{4.1.1}$$

($a < 0$ and $b < 0$ correspond to walking "backwards" relative to the directions of $\mathbf{v}$ and $\mathbf{w}$ respectively). This says that if we start with the distinct directions $\mathbf{v}$ and $\mathbf{w}$ emanating from $\mathbf{0}$, and move along those two directions forwards or backwards however much (or little) we wish, we sweep out exactly the plane $\mathcal{P}$. The description as vectors $a\mathbf{v} + b\mathbf{w}$ for varying scalars a and b is a way of encoding the flatness of $\mathcal{P}$ with two degrees of freedom. Here are some more planes with such a description:

Example 4.1.1. Figures 4.1.2 and 4.1.3 show two other planes in $\mathbf{R}^3$ through $\mathbf{0}$ obtained in a similar way, using $\mathbf{v} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, and using $\mathbf{v}' = \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix}$ and $\mathbf{w}' = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$.

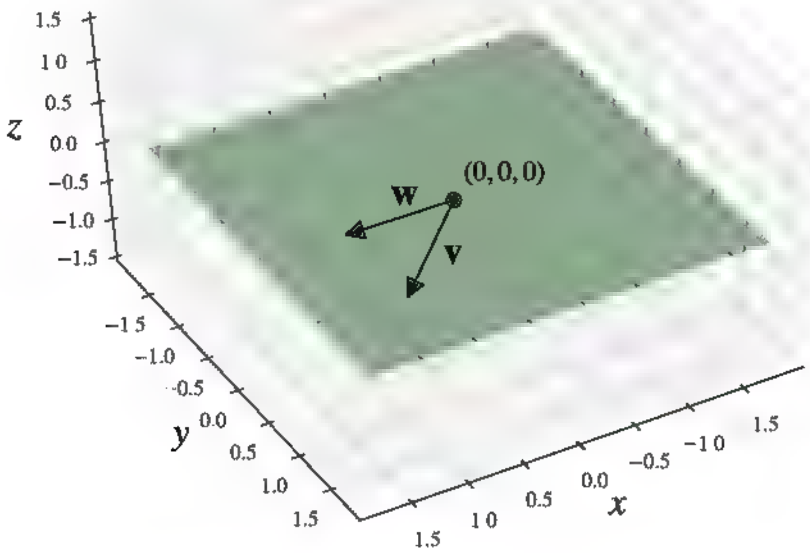


FIGURE 4.1.2. linear combinations of $\mathbf{v}$ and $\mathbf{w}$ sweep out the xy-plane: $z = 0$

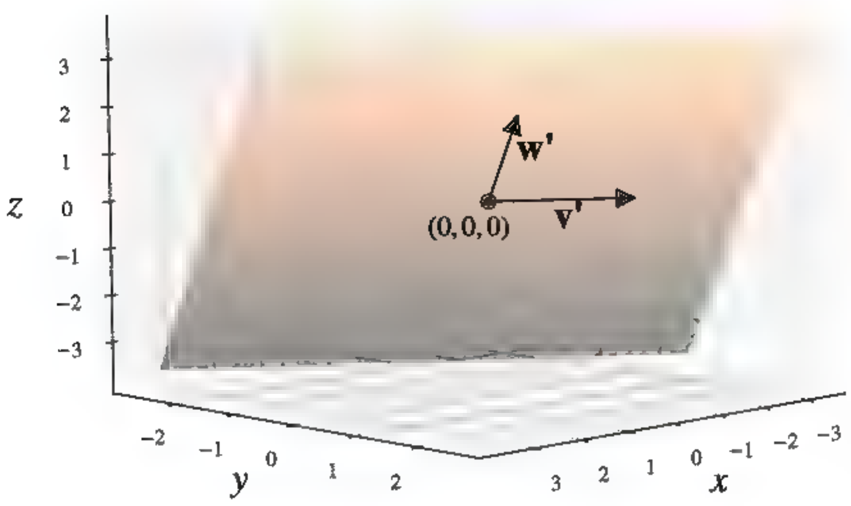


FIGURE 4.1.3. linear combinations of $\mathbf{v}'$ and $\mathbf{w}'$ sweep out a plane

■

Let's revisit the right side of (4.1.1), for $\mathcal{P}$ as in Figure 4.1.1. In addition to forming $a\mathbf{v} + b\mathbf{w}$, what if we keep iterating the vector operations of addition and scalar multiplication? For example, in this way we can obtain $2\mathbf{v} - 3\mathbf{w} + 7\mathbf{v} + 11\mathbf{w}$. However, note that we can group together the $\mathbf{v}$'s and the $\mathbf{w}$'s:

$$2\mathbf{v} - 3\mathbf{w} + 7\mathbf{v} + 11\mathbf{w} = (2+7)\mathbf{v} + (-3+11)\mathbf{w} = 9\mathbf{v} + 8\mathbf{w}.$$

In other words, *any* vector that we can obtain from $\mathbf{v}$ and $\mathbf{w}$ repeatedly using the vector operations (addition and scalar multiplication) in any order is actually of the form $a\mathbf{v} + b\mathbf{w}$ for some scalars a, b.

Thus, the right side of (4.1.1) gives a *parametric form* of the plane through the 3 points $\mathbf{0}, \mathbf{v}, \mathbf{w}$ and describes all vectors created from $\mathbf{v}, \mathbf{w}$ using vector operations. If we instead allow the nonzero 3-vectors $\mathbf{v}, \mathbf{w}$ to lie on a common line through $\mathbf{0}$, which is to say $\mathbf{w}$ is a scalar multiple of $\mathbf{v}$, then the right side of (4.1.1) describes a *line* through $\mathbf{0}$ rather than a plane. Here is an example:

Example 4.1.2. For $\mathbf{v}'' = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$ and $\mathbf{w}'' = \begin{bmatrix} 2 \\ 2 \\ -2 \end{bmatrix} = 2\mathbf{v}''$, the linear combinations of $\mathbf{v}''$ and $\mathbf{w}''$ constitute a line through $\mathbf{0}$ as shown in Figure 4.1.4.

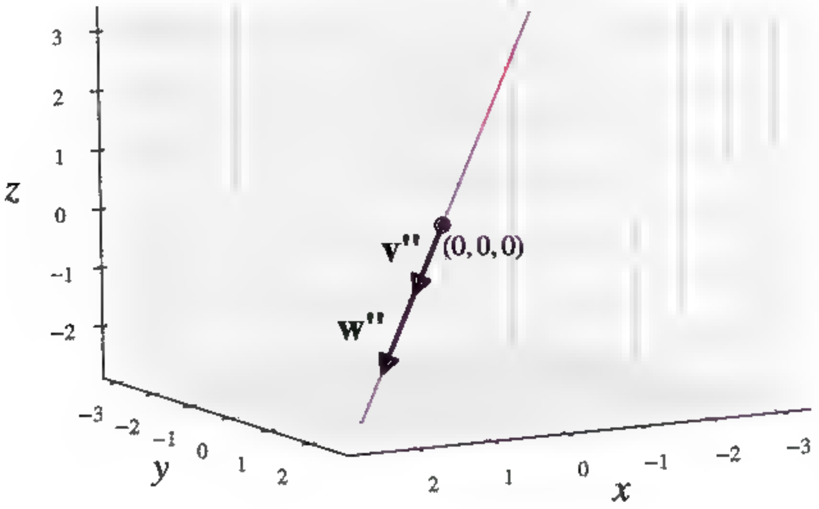


FIGURE 4.1.4. linear combinations of $\mathbf{v}''$ and $\mathbf{w}''$ form a line (not a plane)

■

The preceding visual considerations are special cases of the following very important concept in $\mathbf{R}^n$ for any n.

Definition 4.1.3. The *span* of vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ in $\mathbf{R}^n$ is the collection of all vectors in $\mathbf{R}^n$ that one can obtain from $\mathbf{v}_1, \ldots, \mathbf{v}_k$ by repeatedly using addition and scalar multiplication. In symbols,

$$\mathrm{span}(\mathbf{v}_1, \ldots, \mathbf{v}_k) = \{\text{all } n\text{-vectors } \mathbf{x} \text{ of the form } c_1\mathbf{v}_1 + \cdots + c_k\mathbf{v}_k\}$$

where $c_1, \ldots, c_k$ are arbitrary scalars.

In $\mathbf{R}^3$, for $k = 2$ and nonzero $\mathbf{v}_1, \mathbf{v}_2$ not multiples of each other, this recovers the parametric form (3.2.3) of a plane through $P = \mathbf{0}$. In general, the span of a collection of finitely many n-vectors is the collection of *all* the n-vectors one can reach from those given n-vectors by forming linear combinations in every possible way.

> This is a very new kind of concept – considering such a collection of n-vectors all at the same time. But it is ultimately no different than how we may visualize a plane in our head yet it consists of a lot of different points. **The span of two nonzero n-vectors that are not scalar multiples of each other should be visualized as a "plane" through $\mathbf{0}$ in $\mathbf{R}^n$**; with practice you'll get accustomed to thinking about general spans as an extension of that visualization to larger collections of n-vectors.

Example 4.1.4. Consider the span of one nonzero vector in $\mathbf{R}^2$: if $\mathbf{v} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ then $\mathrm{span}(\mathbf{v})$ is the set of all scalar multiples $c\mathbf{v} = \begin{bmatrix} c \\ 2c \end{bmatrix}$ as in Figure 4.1.5. This is the line through $(0,0)$ with equation $y = 2x$.

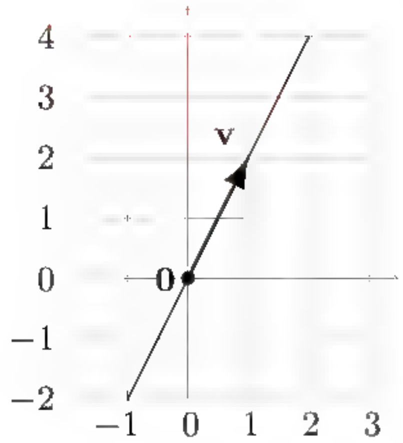


FIGURE 4.1.5. The span of a single nonzero vector is a line through the origin

■

Example 4.1.5. Figures 4.1.2, 4.1.3, and 4.1.4 respectively show: the span of $\mathbf{v} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, the span of $\mathbf{v}' = \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix}$ and $\mathbf{w}' = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$, and the span of $\mathbf{v}'' = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$ and $\mathbf{w}'' = \begin{bmatrix} 2 \\ 2 \\ -2 \end{bmatrix}$.

These illustrate that, in $\mathbf{R}^3$, *the span of nonzero vectors* $\mathbf{v}$ *and* $\mathbf{w}$ *can be either a line or a plane through* $\mathbf{0}$. Namely:

- if $\mathbf{v}$ and $\mathbf{w}$ point in exactly the same direction, or in exactly opposite directions, their span is a line through $\mathbf{0}$;
- otherwise, their span is a plane through $\mathbf{0}$.

■

Example 4.1.6. Let's show that the set U of 4-vectors $\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$ that are perpendicular to $\mathbf{v} = \begin{bmatrix} 2 \\ 3 \\ 1 \\ 7 \end{bmatrix}$ is a span of three 4-vectors. (The same type of calculation as what we are about to do shows that the set of vectors in $\mathbf{R}^n$ perpendicular to *any* fixed nonzero vector in $\mathbf{R}^n$ is a span of $n-1$ nonzero n-vectors.) This is a "higher-dimensional" analogue of our visual experience in $\mathbf{R}^3$ that the collection of vectors in $\mathbf{R}^3$ perpendicular to a given nonzero vector is a plane through the origin (and hence is the span of two nonzero vectors in $\mathbf{R}^3$).

As a first step, we write out the condition of perpendicularity using the dot product:

$$2x + 3y + z + 7w = 0.$$

Now we solve for w (say) to get $w = -(2/7)x - (3/7)y - z/7$. Thus, points of U are precisely

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ -(2/7)x - (3/7)y - z/7 \end{bmatrix} = x \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ -2/7 \end{bmatrix}}_{\mathbf{a}} + y \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \\ -3/7 \end{bmatrix}}_{\mathbf{b}} + z \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \\ -1/7 \end{bmatrix}}_{\mathbf{c}}$$

with *arbitrary* scalars x, y, z. This shows that the vectors perpendicular to $\mathbf{v}$ are precisely those in the span of the vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$ that appear on the right side. (As a safety check on our numerical work, you can check directly that $\mathbf{a}, \mathbf{b}, \mathbf{c}$ really are perpendicular to $\mathbf{v}$ by showing their dot product with this vector equals 0.) This shows that U is the span of $\mathbf{a}, \mathbf{b}, \mathbf{c}$ in $\mathbf{R}^4$.

There is nothing sacred about "solving for w" above. We could just as well have "solved for x" for example, but then we would have obtained a different triple of vectors $\mathbf{a}', \mathbf{b}', \mathbf{c}'$ (whose span is also equal to U). Indeed, the equation of the plane can be rewritten by isolating x on one side as:

$$x = -(3/2)y - z/2 - (7/2)w.$$

Hence, we obtain a rather different-looking description of points in U as

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} -(3/2)y - z/2 - (7/2)w \\ y \\ z \\ w \end{bmatrix} = y \underbrace{\begin{bmatrix} -3/2 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{a}'} + z \underbrace{\begin{bmatrix} -1/2 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{\mathbf{b}'} + w \underbrace{\begin{bmatrix} -7/2 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{\mathbf{c}'}$$

with *arbitrary* scalars y, z, w. As in the preceding calculation, each of $\mathbf{a}', \mathbf{b}', \mathbf{c}'$ is itself also perpendicular to $\mathbf{v}$ and these have span equal to U. ■

We have noted that a line in $\mathbf{R}^2$ or $\mathbf{R}^3$ passing through $\mathbf{0}$ and a plane in $\mathbf{R}^3$ passing through $\mathbf{0}$ each arise as a span of one or two vectors. But lines and planes *not* passing through $\mathbf{0}$ are **not** a span of any collection of vectors! The reason is that the span of any collection of n-vectors **always contains 0**, by setting all coefficients $c_1, \dots, c_k$ in Definition 4.1.3 to be 0, since $0\mathbf{v}_1 + 0\mathbf{v}_2 + \cdots + 0\mathbf{v}_k = \mathbf{0}$.

If V is the span of some finite collection of vectors in $\mathbf{R}^n$ then there are generally *many* different collections of vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$ in $\mathbf{R}^n$ with span equal to V. For example, given one parametric form of a plane in $\mathbf{R}^3$ through $\mathbf{0}$ we get lots of others by rotating the resulting parallelogram grid around the origin in that plane by any nonzero angle we like. To refer to a span while suppressing the mention of any *specific choice* of vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$ that create the span, some new terminology is convenient:

Definition 4.1.7. A *linear subspace* of $\mathbf{R}^n$ is a subset of $\mathbf{R}^n$ that is the span of a finite collection of vectors in $\mathbf{R}^n$. (This is also referred to as a *subspace*, dropping the word "linear".) If V is a linear subspace of $\mathbf{R}^n$, a *spanning set* for V is a collection of n-vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$ whose span equals V. (A linear subspace has **lots** of spanning sets, akin to tiling a floor by parallelogram tiles in many ways.)

Planes and lines in $\mathbf{R}^3$ passing through $\mathbf{0}$ are the visual examples to keep in mind when you hear the phrase "linear subspace". You may wonder: **what is the difference between a linear subspace and a span**? There is no difference, but saying "span" emphasizes the input – a specific finite list $\mathbf{v}_1, \dots, \mathbf{v}_k$ and the dynamic process of forming their linear combinations – whereas saying "linear subspace" emphasizes the output collection V of n-vectors without choosing a specific $\mathbf{v}_1, \dots, \mathbf{v}_k$ whose span is V. It is far more important to know that V can be obtained as a span of *some* list $\mathbf{v}_1, \dots, \mathbf{v}_k$ rather than to pick a *specific* such list. For instance, U in Example 4.1.6 was defined via an orthogonality condition (no mention of spans!) and was then seen to be the span of a couple of different triples of 4-vectors, with neither triple any better than the other. A recurring theme later in this book will be that in many situations a collection of vectors defined without mention of spans turns out to be a linear subspace (as in Example 4.1.6).

Remark 4.1.8. Many students are initially bothered by the fact that a line in $\mathbf{R}^2$ or $\mathbf{R}^3$ not through $\mathbf{0}$ and a plane in $\mathbf{R}^3$ not through $\mathbf{0}$ are *not* linear subspaces according to Definition 4.1.7 (since we have seen that they cannot be the span of a collection of vectors). There is a more general concept than "linear subspace" (called "affine subspace") that allows for situations such as lines and planes in $\mathbf{R}^3$ not passing through $\mathbf{0}$ (e.g., those we encountered in the discussion near the end of Section 3.2); see Definition 4.1.14. But experience in very many situations throughout mathematics and its applications in economics, computer science, natural sciences, and so on shows that **the really key notion is that of linear subspace as defined above**. So don't get worried about the fact that a line or plane in $\mathbf{R}^3$ not passing through $\mathbf{0}$ is not a linear subspace of $\mathbf{R}^3$. If you remain skeptical, experience should change your mind by the end of the course.

Informally, the linear subspace U in $\mathbf{R}^4$ in Example 4.1.6 may seem to be "3-dimensional" since it is presented as a span of three vectors (in $\mathbf{R}^4$). This may even feel geometrically reasonable since U is defined by 1 equation on $\mathbf{R}^4$ and each equation should reduce the "dimension" by 1. Of course, this is just an informal idea since we have given no real definition of what "dimension" means. In Section 4.2 we will make the informal idea of dimension precise (and useful in a much wider context).

You should regard "linear subspace" as serving a role in the geometry of $\mathbf{R}^n$ akin to the fundamental role of lines in Euclidean plane geometry. It was noted early in the Introduction that geometric reasoning in million-dimensional spaces has been an essential ingredient in important improvements in MRI technology, and we next discuss a way that linear subspaces arise in economics and neuroscience.

Example 4.1.9. In *modern portfolio theory* (MPT), a blend of investments among N specified assets is designed to minimize financial risk subject to a desired overall expected rate of return. It turns out that there is a unique solution to this problem (computed in terms of probabilistic data determined from historical performance). The initial ideas in this direction were worked out by Harry Markowitz in the early 1950's while he was a graduate student and earned him the 1990 Nobel Prize in economics.

This unique solution is given by a formula usually obtained from N-variable optimization techniques (such as Lagrange multipliers, the focus of Chapter 12; see [LX, (3.9)-(3.12)]). However, the formula can be understood using just linear algebra in $\mathbf{R}^N$, based on the following features of linear subspaces to be developed later in this book. For a plane $\mathcal{P}$ through the origin in $\mathbf{R}^3$, and a point $\mathbf{w}$ not in the plane, visual experience shows that there is a single point in $\mathcal{P}$ closest to $\mathbf{w}$. There is an analogue for linear subspaces V of any $\mathbf{R}^n$: if $\mathbf{w}$ is a point in $\mathbf{R}^n$ outside V then there is exactly one point $\mathbf{v}$ in V (called the "projection" of $\mathbf{w}$ into V) for which the distance $\|\mathbf{v}-\mathbf{w}\|$ is minimal, as will be discussed in detail in Chapters 6 and 19 (see Theorem 6.2.1). Moreover, just as the parametric form of a plane in $\mathbf{R}^3$ expresses the 2-dimensionality of planes, for general linear subspaces V in any $\mathbf{R}^n$ soon we will define a concept of "dimension" for V (see Definition 4.2.4) based on spanning sets.

Coming back to economics, the unique solution to the risk-minimizing task in MPT for N assets can be interpreted via distance-minimization from a point in $\mathbf{R}^N$ to a specific $(N-2)$-dimensional linear subspace in $\mathbf{R}^N$ (see [RPGS], [KK, Sec. 8] for details). In particular, the uniqueness of the solution in MPT is an instance of the uniqueness of the closest point to a linear subspace (in Theorem 6.2.1). We'll come back to this at the end of Remark 19.3.9.

The *same* math (distance-minimization to linear subspaces) enables the brain to recognizes faces by using $\mathbf{R}^{50}$ as a "face space" (see [CT], [Daj], [Q], [T]). According to [Daj], ". . . deep in the brain's visual system, the neurons are actually doing simple linear algebra. Each cell is literally taking a 50-dimensional vector space – face space – and projecting it onto a 1-dimensional subspace . . . this completely overturns the long-standing idea that single face cells are coding specific facial identities. Instead, what we've found is that these cells are beautifully simple linear projection machines." We'll return to this in Example 6.2.2. (Linear projection models for sensory recognition in the brain appears to be a universal theme, but nobody knows why. A causal understanding should be obtained some day via optogenetics.) ■

Having given informal indications about two applications, we want to turn to some explicit examples and non-examples of linear subspaces.

Example 4.1.10. Let $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$, $\mathbf{v}_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$, so $\mathrm{span}(\mathbf{v}_1, \mathbf{v}_2)$ consists of all vectors of the form $a\mathbf{v}_1 + b\mathbf{v}_2 = \begin{bmatrix} a+b \\ b \\ 0 \\ 0 \end{bmatrix}$. We can make the second coordinate whatever we like and then (by choosing a suitably) we can make the first coordinate whatever we like. But the third and fourth coordinates vanish.[7] ■

For efficiency of language, from now on we shall write

$$\mathbf{v} \in \mathbf{R}^n$$

as shorthand for "$\mathbf{v}$ belongs to $\mathbf{R}^n$." (The symbol "$\in$" means "element of".) In plain language it says "$\mathbf{v}$ is an n-vector." For instance, in Example 4.1.10 our calculation of the span there says this:

$$\mathrm{span}(\mathbf{v}_1, \mathbf{v}_2) = \left\{ \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \in \mathbf{R}^4 : x_3 = x_4 = 0 \right\}.$$

(See the short table of useful notation in Table 0.0.1 near the end of the Introduction.)

If $\mathbf{x}$ and $\mathbf{y}$ belong to a linear subspace V of $\mathbf{R}^n$ (say V is the span of some vectors $\mathbf{v}_1, \dots, \mathbf{v}_k \in \mathbf{R}^n$) then so does $5\mathbf{x} - 3\mathbf{y}$. Indeed, writing

$$\mathbf{x} = c_1\mathbf{v}_1 + \cdots + c_k\mathbf{v}_k, \quad \mathbf{y} = d_1\mathbf{v}_1 + \cdots + d_k\mathbf{v}_k \in V$$

for some scalars $c_1, \dots, c_k$ and $d_1, \dots, d_k$, we have

$$5\mathbf{x} - 3\mathbf{y} = 5c_1\mathbf{v}_1 + \cdots + 5c_k\mathbf{v}_k - 3d_1\mathbf{v}_1 - \cdots - 3d_k\mathbf{v}_k = (5c_1 - 3d_1)\mathbf{v}_1 + \cdots + (5c_k - 3d_k)\mathbf{v}_k$$

and similarly for $a\mathbf{x} + b\mathbf{y}$ for any scalars a and b in place of 5 and 3 respectively. There is nothing special about using two vectors in a subspace. More generally, the same type of calculation yields:

Proposition 4.1.11. If V is a linear subspace in $\mathbf{R}^n$ then for any vectors $\mathbf{x}_1, \dots, \mathbf{x}_m \in V$ and scalars $a_1, \dots, a_m$ the linear combination $a_1\mathbf{x}_1 + \cdots + a_m\mathbf{x}_m$ also lies in V. In words: all linear combinations of n-vectors chosen from a linear subspace of $\mathbf{R}^n$ belong to that same subspace.

This result is important for two reasons: (i) it provides special algebraic properties of linear subspaces in any $\mathbf{R}^n$ that somehow encode the idea of "flatness", (ii) this good behavior under linear combinations will turn out to *exactly characterize* linear subspaces among general collections of n-vectors (see Theorem

---

[7] In mathematics, the word "vanish" means "is equal to zero": it applies to scalars (the number 0), functions (a function with value 0 everywhere), n-vectors ($\mathbf{0}$), etc.

21.3.7). So Proposition 4.1.11 (alongside its deeper refinement in Theorem 21.3.7) is a bridge for thinking about linear subspaces both geometrically and algebraically. Proposition 4.1.11 fails for any line or plane in $\mathbf{R}^3$ that does not pass through $\mathbf{0}$ (which we have noted are *not* linear subspaces).

Example 4.1.12. Partly as an application of Proposition 4.1.11, for each of Figures 4.1.6–4.1.8 below let's discuss what parts are linear subspaces or not (and why). In Figure 4.1.6, the blue plane P_2 goes through $\mathbf{0}$ and so is a linear subspace. The red plane P_1 does not go through $\mathbf{0}$ and so is not a linear subspace.

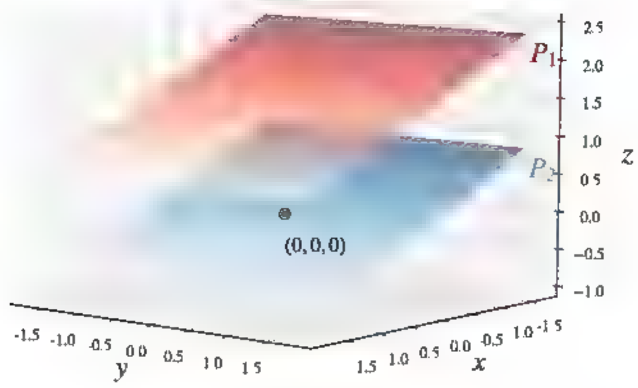


FIGURE 4.1.6. A pair of parallel planes; only the blue one goes through $\mathbf{0}$

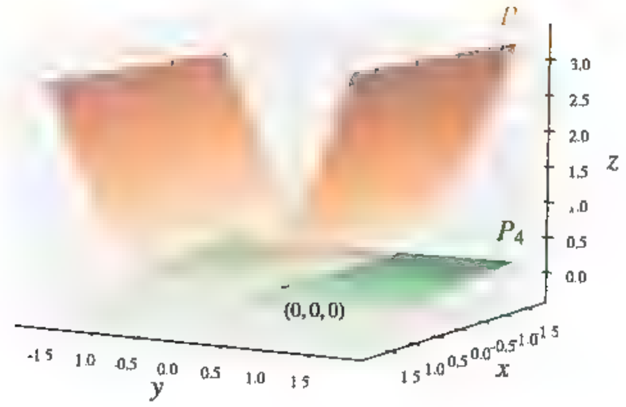


FIGURE 4.1.7. A green plane and an orange "paraboloid", both passing through $\mathbf{0}$

In Figure 4.1.7, the green region P_4 is a plane through $\mathbf{0}$, so it is a linear subspace. The orange "paraboloid" P_3 is not a linear subspace, because it violates the conclusion of Proposition 4.1.11 in many ways. For instance, if we pick a nonzero point on this orange surface then the line joining it to the origin is essentially never inside the surface. Also, if we pick two different points on that orange surface then the segment joining them (consisting of convex linear combinations) is generally not contained in the surface.

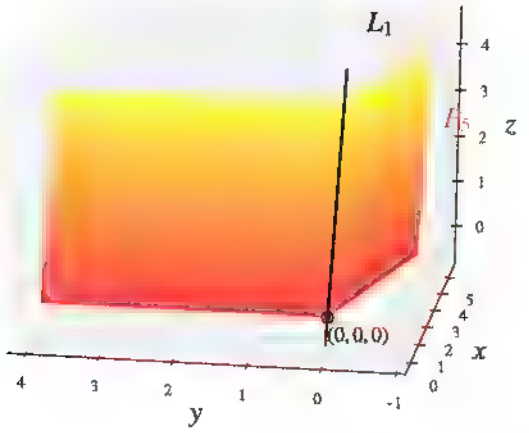


FIGURE 4.1.8. A bent plane, folded along a line L_1 passing through $\mathbf{0}$, is not a linear subspace

Finally, in Figure 4.1.8, pick a point $\mathbf{v}$ on the bent surface P_5 that doesn't lie along the fold L_1. The scalar multiples $t\mathbf{v}$ for $t \geq 0$ lie in the bent surface, but if we consider $t < 0$ then $t\mathbf{v}$ lies "across the line" L_1 and so is outside the bent surface (due to the folding along L_1). Hence, P_5 is *not* a linear subspace (once again, due to violating the conclusion of Proposition 4.1.11). ■

Example 4.1.13. Consider the set W of all vectors in $\mathbf{R}^5$ that are perpendicular to *both* $\begin{bmatrix} 1 \\ 1 \\ 0 \\ -1 \\ 2 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 2 \\ 3 \\ 1 \\ -1 \end{bmatrix}$.
We will now find three explicit 5-vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ so that $W = \text{span}(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$, and in particular W is a linear subspace of $\mathbf{R}^5$. (This phenomenon is not specific to W; it works for perpendicularity against any finite collection of vectors in any $\mathbf{R}^n$. To understand this in a general setting involves relating the algebraic and geometric aspects of linear algebra: showing that the solution set to any such system of conditions is a span of a finite set of vectors. This will be addressed in Section 21.3; see Theorem 21.3.7.)

As we did in Example 4.1.6, write out the two perpendicularity requirements in terms of dot products. Namely, a vector $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$ in $\mathbf{R}^5$ lies in W precisely when it satisfies the two "perpendicularity equations"

$$x_1 + x_2 - x_4 + 2x_5 = 0, \quad 2x_2 + 3x_3 + x_4 - x_5 = 0.$$

We can rewrite these two equations as expressing two of the variables in terms of the others (with no further constraints in these other variables), say x_1 and x_3 in terms of x_2, x_4, and x_5: the pair of equations equivalently says

$$x_1 = -x_2 + x_4 - 2x_5, \quad x_3 = -(2/3)x_2 - (1/3)x_4 + (1/3)x_5.$$

Since there are *no restrictions at all* on x_2, x_4, and x_5, W is the collection of vectors of the form

$$\begin{bmatrix} -x_2 + x_4 - 2x_5 \\ x_2 \\ -(2/3)x_2 - (1/3)x_4 + (1/3)x_5 \\ x_4 \\ x_5 \end{bmatrix}$$

for arbitrary scalars x_2, x_4, x_5. This vector can be expressed in the form of a linear combination by separating out the parts involving each of x_2, x_4, x_5 separately:

$$\begin{bmatrix} -x_2 \\ x_2 \\ -(2/3)x_2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} x_4 \\ 0 \\ -(1/3)x_4 \\ x_4 \\ 0 \end{bmatrix} + \begin{bmatrix} -2x_5 \\ 0 \\ (1/3)x_5 \\ 0 \\ x_5 \end{bmatrix} = x_2 \begin{bmatrix} -1 \\ 1 \\ -2/3 \\ 0 \\ 0 \end{bmatrix} + x_4 \begin{bmatrix} 1 \\ 0 \\ -1/3 \\ 1 \\ 0 \end{bmatrix} + x_5 \begin{bmatrix} -2 \\ 0 \\ 1/3 \\ 0 \\ 1 \end{bmatrix}.$$

In other words, W is the span of the vectors $\mathbf{v}_1 = \begin{bmatrix} -1 \\ 1 \\ -2/3 \\ 0 \\ 0 \end{bmatrix}$, $\mathbf{v}_2 = \begin{bmatrix} 1 \\ 0 \\ -1/3 \\ 1 \\ 0 \end{bmatrix}$, and $\mathbf{v}_3 = \begin{bmatrix} -2 \\ 0 \\ 1/3 \\ 0 \\ 1 \end{bmatrix}$. ■

There is nothing special about describing points of W in Example 4.1.13 in terms of the 3 independently varying parameters x_2, x_4, x_5: that is an artifact of our initial decision to solve the two "perpendicularity" equation for x_1 and x_3 in terms of the other three variables among the 5 given variables. We could have also decided, for example, to solve for x_4 and x_5 in terms of x_1, x_2, x_3. This would yield the description of W as the span of a rather different triple of vectors

$$\mathbf{v}'_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \\ -1 \end{bmatrix}, \quad \mathbf{v}'_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -5 \\ -3 \end{bmatrix}, \quad \mathbf{v}'_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ -8 \\ -3 \end{bmatrix}$$

at the cost of some algebra that we don't want to consider yet (see Chapter 22 for a systematic treatment).

If we had chosen to solve for two other variables in terms of the others in the given pair of equations then we would have arrived at a *rather different collection* of three vectors in $\mathbf{R}^5$ that also span W. There is nothing special about the spanning set of three vectors for W obtained at the end of Example 4.1.13, much as a plane through the origin in $\mathbf{R}^3$ can be written as a span of zillions of different pairs of vectors.

The preceding descriptions of W suggest that it should be "3-dimensional" (as a subspace of $\mathbf{R}^5$), to be justified in Example 5.1.6. Note also that W is defined in $\mathbf{R}^5$ by 2 equations (the orthogonality conditions) and so should have "dimension" $5 - 1 - 1 = 3$ since each equation should reduce the "dimension" by 1.

Lines and planes not passing through $\mathbf{0}$ are instances of a more general concept than linear subspace:

Definition 4.1.14 (optional). An *affine subspace* of $\mathbf{R}^n$ is the set of all vectors in $\mathbf{R}^n$ of the form $\mathbf{a} + \mathbf{v}$ where $\mathbf{v}$ belongs to a fixed linear subspace V of $\mathbf{R}^n$ and $\mathbf{a} \in \mathbf{R}^n$ is a fixed vector.

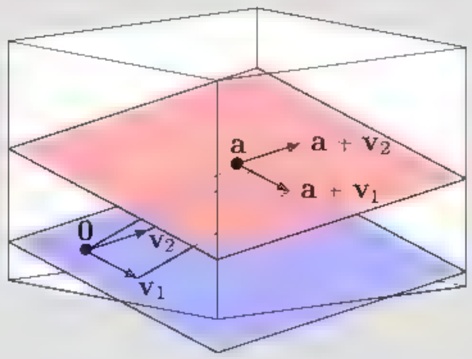


FIGURE 4.1.9. An affine subspace (red) in $\mathbf{R}^3$ that does not pass through $\mathbf{0}$.

Example 4.1.15. Consider the plane $\mathcal{P}$ given by the equation $x - y + z = 2$; this is the red plane in Figure 4.1.9. The point $\mathbf{a} = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}$ lies on this plane. The two vectors $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$ and $\mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ span a linear subspace V in $\mathbf{R}^3$ (this is the linear subspace shown in blue in Figure 4.1.9). One can check that

$$\mathcal{P} = \{\mathbf{a} + \mathbf{v} : \mathbf{v} \in V = \mathrm{span}(\mathbf{v}_1, \mathbf{v}_2)\},$$

and hence $\mathcal{P}$ is an affine subspace of $\mathbf{R}^3$ (but $\mathcal{P}$ is not a linear subspace, since it does not pass through the origin). ■

Similarly, *any* line in $\mathbf{R}^2$ is an affine subspace of $\mathbf{R}^2$ and *any* line or plane in $\mathbf{R}^3$ is an affine subspace of $\mathbf{R}^3$, whether or not they pass through $\mathbf{0}$. From the point of view of geometry, affine subspaces are very natural. There is even an analogue of Proposition 4.1.11 for affine subspaces, requiring the condition $a_1 + a_2 + \cdots + a_m = 1$. For the remainder of this book (and in most applications of linear algebra), it will usually be sufficient just to consider linear subspaces. In particular, whenever we say "subspace" it is always understood to mean "linear subspace", so it passes through $\mathbf{0}$.

4.2. Dimension. We would like to define the "dimension" of a linear subspace in a way that generalizes our familiar concept of dimension: a thread or a line is 1-dimensional, a sheet of paper is 2-dimensional, the world around us is 3-dimensional. To motivate where we are going, let us first try to put into more precise language the intuition that says a thread is 1-dimensional whereas a sheet of paper is 2-dimensional. If an ant crawls along a thread, we only need one measurement to tell us where it is: the distance to the end of the thread. But if an ant crawls on a sheet of paper, we need two measurements: how far it is from the left side of the paper and how far it is from the bottom side.

Example 4.2.1. In order to identify the occurrence of a physical event unambiguously, we need to specify the location (x, y, z) in space where it occurs *and* a time t at which it occurs (relative to some reference time, so $t < 0$ corresponds to occurring before the reference time). Putting these together, we conclude that 4-vectors (x, y, z, t) are the complete information to specify where and when an event occurs.

The collection of all such 4-vectors is $\mathbf{R}^4$, but when interpreted physically via the 3 coordinates in space and 1 coordinate in time it is called *spacetime*. This is the framework implicitly used throughout the discussion of many scientific phenomena in physics, and it is 4-dimensional in the sense that a "point" of spacetime involves the specification of 4 independent numbers (3 for space, 1 for time).

If we consider the region in spacetime with a *fixed* value t_0 of time, we get the collection of points (x, y, z, t_0) in which only x, y, z vary. This recovers the familiar fact that at a given time, to specify an event involves a 3-dimensional range of possibilities (coordinates in space). One of Einstein's key discoveries in his Theory of Relativity is that when a particle moves through spacetime under the influence of gravity, its 4 spacetime coordinates along its path of motion are linked in a rather subtle manner (whose precise formulation lies beyond the level of this course).

Keep in mind that the preceding is *just one* application of $\mathbf{R}^4$. In our study of correlation, we saw that measuring if 4 data points in $\mathbf{R}^2$ are "well-approximated" by a line is understood in terms of vectors in $\mathbf{R}^4$! Thus, how to "interpret" $\mathbf{R}^n$ is really a matter of any particular application one has at hand. There is no special all-powerful interpretation of $\mathbf{R}^4$ or any $\mathbf{R}^n$ that is more important than any other; everything depends on the purpose one has for a given mathematical problem. ■

Example 4.2.2. As a "non-physical" example of higher dimensions that you have worked with a lot in algebra (perhaps without realizing it), consider the collection P_4 of all polynomials of degree at most 4.

Such a polynomial has the form $ax^4 + bx^3 + cx^2 + dx + e$ where a, b, c, d, e can be any real numbers. This description involves 5 "independent" choices (namely, the choices of a, b, c, d, e). So in a sense that can be made precise (but we don't dwell on in this book), P_4 is 5-dimensional. ■

In particular, we may informally say:

> the "dimension" of an object X tells us how many different numbers are needed to locate a point in X.

To turn this into something unambiguous and *useful*, we now focus on the case of a linear subspace V of $\mathbf{R}^n$, where vector algebra will provide a way to make that informal idea precise. The "dimension" of V will be, intuitively, the number of independent directions in V. In other words, it will tell us how many numbers we need in order to specify a vector $\mathbf{v}$ in V.

More precisely, recall that by the definition of "linear subspace" (see Definition 4.1.7), V is the span of some finite collection of vectors $\mathbf{v}_1, \dots, \mathbf{v}_k \in \mathbf{R}^n$. That is, for any $\mathbf{v} \in V$ we can write

$$\mathbf{v} = c_1\mathbf{v}_1 + \cdots + c_k\mathbf{v}_k$$

for some scalars $c_1, \dots, c_k$, so to determine $\mathbf{v}$ it is enough to tell us k numbers – the scalars $c_1, \dots, c_k$. But V can have another spanning set consisting of a *different* number of vectors, as we now illustrate.

Example 4.2.3. The span V of two nonzero vectors in $\mathbf{R}^3$ could be a line (such as if the two vectors point in the same or opposite directions), in which case V is also spanned by just one of those two vectors (e.g., the second vector is redundant). For instance, if $\mathbf{v} = \begin{bmatrix} 2 \\ -2 \\ 0 \end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix} -3 \\ 3 \\ 0 \end{bmatrix}$ then $\mathbf{w} = -(3/2)\mathbf{v}$ and so $\mathrm{span}(\mathbf{v}, \mathbf{w}) = \mathrm{span}(\mathbf{v})$ (since $a\mathbf{v} + b\mathbf{w} = a\mathbf{v} + b(-3/2)\mathbf{v} = (a - 3b/2)\mathbf{v}$).

Similarly, the span of three nonzero vectors in $\mathbf{R}^3$ could be a plane in special circumstances (or even a line in especially degenerate circumstances). For instance, if $\mathbf{v} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$, $\mathbf{v}' = \begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix}$, $\mathbf{v}'' = \begin{bmatrix} 2 \\ -1 \\ -1 \end{bmatrix}$ then $\mathbf{v}'' = 2\mathbf{v} - \mathbf{v}'$ by inspection, so $\mathrm{span}(\mathbf{v}, \mathbf{v}', \mathbf{v}'') = \mathrm{span}(\mathbf{v}, \mathbf{v}')$ because

$$a\mathbf{v} + a'\mathbf{v}' + a''\mathbf{v}'' = a\mathbf{v} + a'\mathbf{v}' + a''(2\mathbf{v} - \mathbf{v}') = (a + 2a'')\mathbf{v} + (a' - a'')\mathbf{v}'.$$

In both such cases, the initial spanning set has some redundancy. To define "dimension" for V, we want to use ways of spanning V that (in a sense we need to make precise) *don't have redundancy*. ■

Definition 4.2.4. Let V be a nonzero linear subspace of some $\mathbf{R}^n$. The *dimension* of V, denoted as $\dim(V)$, is defined to be the smallest number of vectors needed to span V. We define $\dim(\{\mathbf{0}\}) = 0$.

Theorem 4.2.5. For $k \geq 2$, consider a collection $\mathbf{v}_1, \dots, \mathbf{v}_k$ of vectors spanning a linear subspace V in $\mathbf{R}^n$. We have $\dim(V) = k$ precisely when "there is no redundancy": *each $\mathbf{v}_i$ is not a linear combination of the others*, or in other words removing it from the list destroys the spanning property.

Equivalently, $\dim(V) < k$ precisely when "there is redundancy": *some $\mathbf{v}_i$ is a linear combination of the others*, or in other words removing some $\mathbf{v}_i$ from the list does *not* affect the span.

(If some $\mathbf{v}_i$ vanishes then it is a linear combination of the others and hence can be dropped from the span, so $\dim(V) < k$. Thus, the essential case in this result is when all $\mathbf{v}_i$ are nonzero. A more general formulation is given in Chapter 19 as Theorem 19.2.3, which is proved in Section B.1.)

Example 4.2.6. Consider the 3-vectors $\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, $\mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$, $\mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$. These span $\mathbf{R}^3$ because

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = a \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + b \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + c \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = a\mathbf{e}_1 + b\mathbf{e}_2 + c\mathbf{e}_3$$

yet no two of the $\mathbf{e}_i$'s span $\mathbf{R}^3$ (e.g., anything in the span of $\mathbf{e}_1$ and $\mathbf{e}_3$ can't have nonzero second entry). Hence, by Theorem 4.2.5, $\dim \mathbf{R}^3 = 3$. ■

As a further illustration of Theorem 4.2.5, if V is a nonzero linear subspace of $\mathbf{R}^3$ then we have the following possibilities (illustrated in Figure 4.2.1).

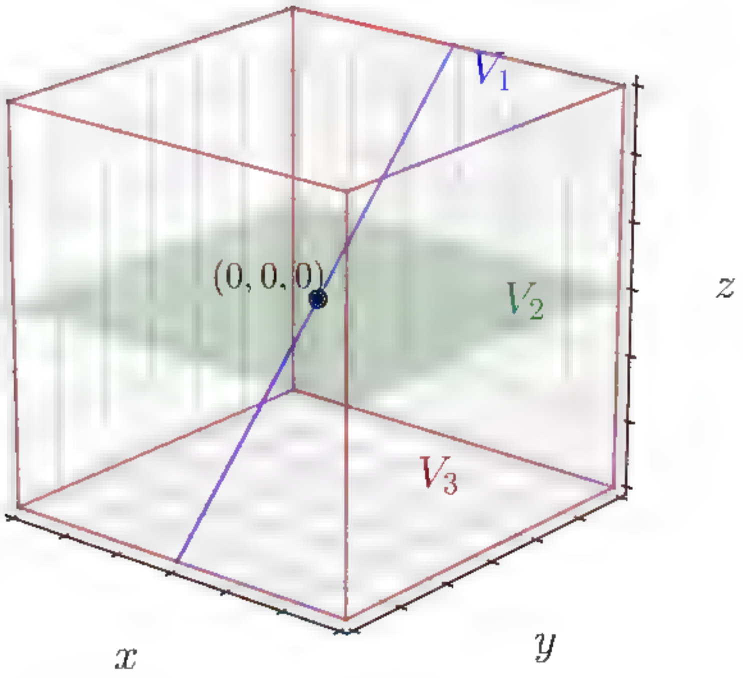


FIGURE 4.2.1. Linear subspaces V_i in $\mathbf{R}^3$ with $\dim V_i = i$ ($V_3 = \mathbf{R}^3$)

- $\dim(V_1) = 1$: this means V_1 is a line passing through $(0,0,0)$.
- $\dim(V_2) = 2$: this means V_2 is a plane passing through $(0,0,0)$.
- $\dim(V_3) = 3$: this is possible too. It means $V_3 = \mathbf{R}^3$. (It is geometrically plausible that under the preceding definition of dimension the only 3-dimensional subspace of $\mathbf{R}^3$ is itself, and that any subspace of $\mathbf{R}^3$ has dimension at most 3. A systematic way to compute the dimension of spans of at most three vectors will be given in Section 5.1. More general $\mathbf{R}^n$-analogues based on the preceding definition of dimension are provided in Chapter 19 for those who are interested.)

Remark 4.2.7 (online resource). In the video series "Essence of Linear Algebra" mentioned in Remark 1.3.7, the second video provides visualizations for the concepts of linear combination, span, and dimension, as well as the "redundancy" aspect of Theorem 4.2.5.

Please always remember the "Alert" in Remark 1.3.7 whenever consulting online resources that you find on your own. This is also the reason that we shall refer you only to specific videos in the series "Essence of Linear Algebra".

Here is a geometrically plausible property of dimension (which is proved in Section B.1).

Theorem 4.2.8. If V and W are linear subspaces of $\mathbf{R}^n$ with W contained in V (i.e., every vector in W also belongs to V, much like a line inside a plane in $\mathbf{R}^3$) then $\dim W \leq \dim V$, and equality holds precisely when $W = V$.

How do we figure out if there is redundancy in a collection of nonzero vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k \in \mathbf{R}^n$ spanning a linear subspace V of $\mathbf{R}^n$? In general, we cannot figure this out by just staring at the $\mathbf{v}_i$'s. For instance:

Example 4.2.9. Is the span of

$$\begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \\ -1 \end{bmatrix}, \begin{bmatrix} 2 \\ -4 \\ 3 \end{bmatrix}$$

actually the entirety of $\mathbf{R}^3$? The answer is "no" – these vectors all belong to a common plane *through the origin* in $\mathbf{R}^3$ – but this isn't evident by inspection. We need to do some work to show that one of these vectors (in fact, any one of them) belongs to the span of the other two. We will return to this in Example 5.1.7. ■

Example 4.2.10. Is the linear subspace W in $\mathbf{R}^5$ considered in Example 4.1.13 actually 3-dimensional? There it was seen that W is the span of 3 explicit vectors in $\mathbf{R}^5$, but how do we know it can't also be written as a span of fewer vectors? We will come back to this in Example 5.1.6 (and revisit it by other methods later on). ■

Chapter 4 highlights (links to highlights in and next chapters)

Notation	**Meaning**	**Location in text**
$\mathrm{span}(\mathbf{v}_1, \dots, \mathbf{v}_k)$	the span of a collection of n-vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$	Definition 4.1.3
$\in$ (e.g., "$\mathbf{v} \in \mathbf{R}^n$")	element of (or: belonging to)	box below Example 4.1.10
$\dim V$ (or $\dim(V)$)	dimension of a linear subspace V in $\mathbf{R}^n$	Definition 4.2.4

Concept	**Meaning**	**Location in text**
span	the collection of *all* linear combinations of given n-vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$	Definition 4.1.3
linear subspace (of $\mathbf{R}^n$)	the span of a list of n-vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$	Definition 4.1.7
dimension	smallest number of vectors that span a given linear subspace V in $\mathbf{R}^n$ (with $\dim(\{\mathbf{0}\}) = 0$ by definition)	Definition 4.2.4

Result	**Meaning**	**Location in text**
linear subspaces are preserved under linear combinations, and *always* contain $\mathbf{0}$	for a linear subspace V of $\mathbf{R}^n$ and any $\mathbf{v}_1, \dots, \mathbf{v}_k \in V$, we have $c_1\mathbf{v}_1 + \cdots + c_k\mathbf{v}_k \in V$ for all scalars $c_1, \dots, c_k$ (in particular, $\mathbf{0} \in V$ by choosing $c_1 = 0, \dots, c_k = 0$)	Proposition 4.1.11
dimension encodes redundancy of a span	for n-vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$ with $k \geq 2$, $\dim \mathrm{span}(\mathbf{v}_1, \dots, \mathbf{v}_k) < k$ precisely when some $\mathbf{v}_j$ is a linear combination of the others (so omitting $\mathbf{v}_j$ yields the same span)	Theorem 4.2.5
a containment of linear subspaces is an equality precisely when they have same dimension	if W and V are linear subspaces of $\mathbf{R}^n$ with W contained in V then $W \neq V$ precisely when $\dim W < \dim V$ (so $W = V$ precisely when $\dim W = \dim V$)	Theorem 4.2.8

Skill	**Location in text**
express orthogonality to one or two n-vectors as a span	Examples 4.1.6 and 4.1.13
recognize when a picture of a collection of vectors in $\mathbf{R}^3$ (or $\mathbf{R}^2$) is a linear subspace	Example 4.1.12
understand that a (nonzero) linear subspace of $\mathbf{R}^n$ can be described as a span of *many* different collections of n-vectors	Examples 4.1.6 and 4.1.13
apply the *definition* of dimension (of linear subspace) to lines and planes through $\mathbf{0}$ in $\mathbf{R}^3$, as well as to spans or one or two vectors in $\mathbf{R}^3$	Example 4.2.3 and Figure 4.2.1 (and the list of cases just below it)

5. Basis and orthogonality

In the previous chapter we encountered the new concepts of linear subspace of $\mathbf{R}^n$ and dimension thereof (extending the concepts of line and plane through the origin in $\mathbf{R}^3$). We also recognized that despite the geometric appeal of the definition of dimension (as the minimal number of vectors whose span is a given linear subspace), it isn't immediately apparent how to actually compute the dimension of a given linear subspace (say, when it is given as a span of an explicit finite collection of n-vectors). We shall now introduce the important notion of "basis" for a linear subspace that will help us to organize our discussion for how to compute dimensions in some situations. A special case of this, called an "orthogonal basis", will be especially useful in our subsequent development of ideas in linear algebra.

By the end of this chapter, you should be able to:

- determine a basis, and dimension, for a linear subspace of $\mathbf{R}^2$ and $\mathbf{R}^3$ (Chapter 19 treats general $\mathbf{R}^n$);
- verify whether a collection of vectors in $\mathbf{R}^n$ is orthogonal or orthonormal.

5.1. Basis and dimension computations.

Definition 5.1.1. A *basis* for a nonzero linear subspace V in $\mathbf{R}^n$ is a spanning set for V consisting of exactly $\dim(V)$ vectors.

Example 5.1.2. If $\dim(V) = 2$ then a basis for V consists of any $\mathbf{v}, \mathbf{w}$ for which $\mathrm{span}(\mathbf{v}, \mathbf{w}) = V$. ■

Example 5.1.3. By Example 4.2.6, one basis of $\mathbf{R}^3$ is given by $\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$; this is called the *standard basis* of $\mathbf{R}^3$. But *many other triples* of vectors are also a basis of $\mathbf{R}^3$ (an explicit additional triple will be given later in (5.3.2)). Something all bases have in common is that they yield a "tiling" of space by a grid of parallelotopes, as shown in Figure 5.1.1.

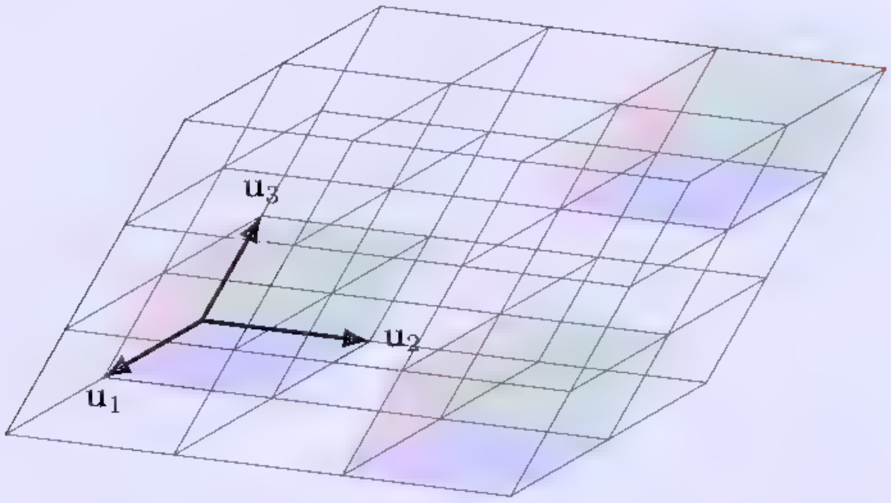


FIGURE 5.1.1. A typical basis of $\mathbf{R}^3$ and resulting "parallelotope grid" for space

A basis of a plane through $\mathbf{0}$ in $\mathbf{R}^3$ (resp. a basis of $\mathbf{R}^3$) yields a "grid" in the plane (resp. in space) telling us where any point is (uniquely) located relative to the grid. **The non-orthogonal $\mathbf{u}_j$'s in Figure 5.1.1 (and not the "standard basis") are the best mental image for a "typical" basis of $\mathbf{R}^3$.** ■

Example 5.1.4. For each pair of nonzero vectors in Figure 5.1.2, let's discuss if it is a basis for $\mathbf{R}^2$ or not.

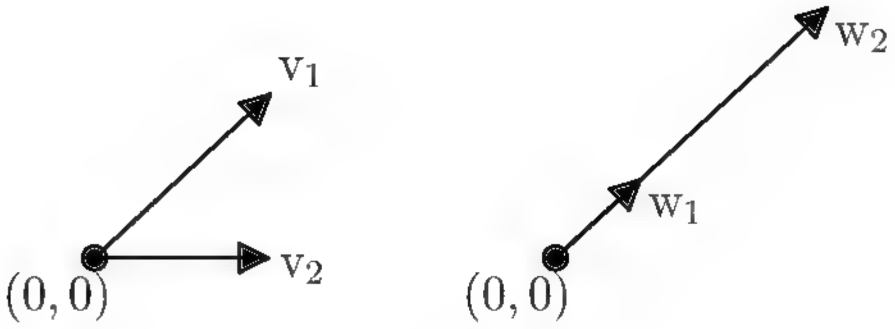


FIGURE 5.1.2. Two pairs of vectors in $\mathbf{R}^2$, one a basis and one not

The vectors $\mathbf{w}_1, \mathbf{w}_2$ as shown lie on the same line through the origin, so their span (or even the span of just one of them) is that line. Hence, they do not span $\mathbf{R}^2$ and so are not a basis.

On the other hand, we claim that $\mathbf{v}_1, \mathbf{v}_2$ do span $\mathbf{R}^2$. To explain this geometrically, the key observation is that these vectors generate a "parallelogram grid" tiling the plane as shown in Figure 5.1.3. The corners of the small blue parallelograms are $n\mathbf{v}_1 + m\mathbf{v}_2$ for integers n and m (possibly negative): beginning at $\mathbf{0}$ we move backwards or forwards some number of times along the $\mathbf{v}_1$-direction and $\mathbf{v}_2$-direction to reach each corner; Figure 5.1.3 shows that $2\mathbf{v}_1 - \mathbf{v}_2$ is 2 steps forward along $\mathbf{v}_1$ and 1 step backwards along $\mathbf{v}_2$.

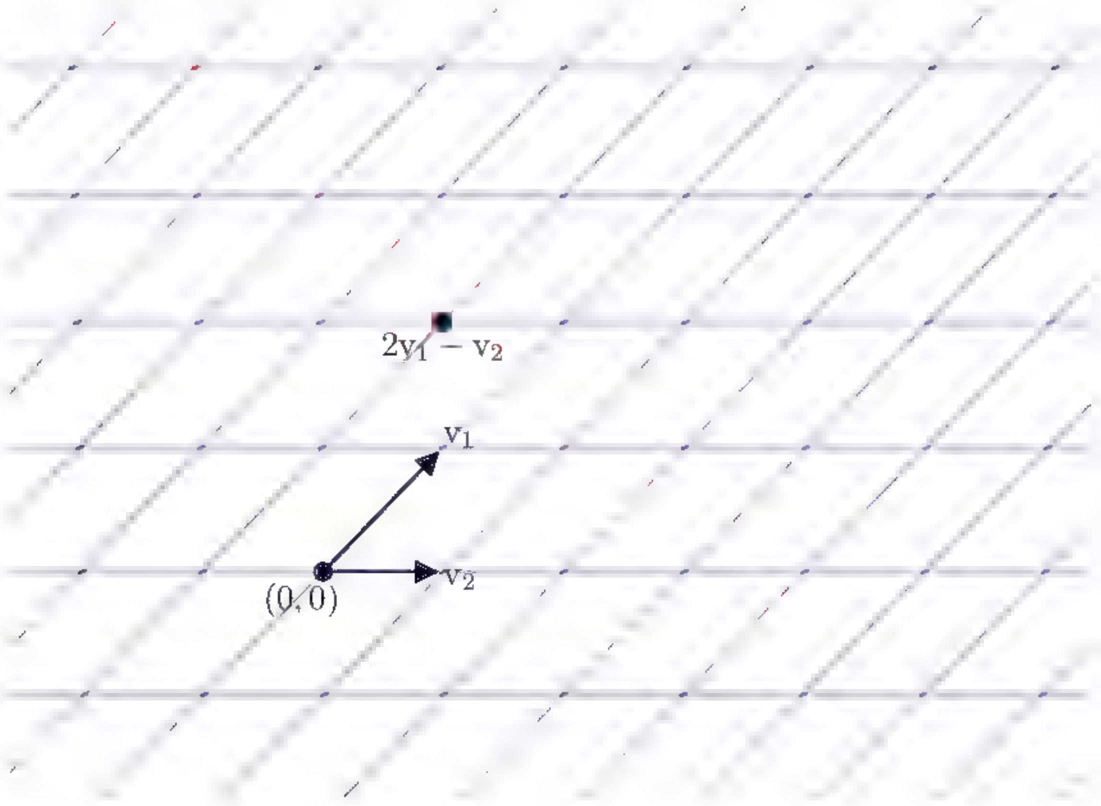


FIGURE 5.1.3. Parallelogram grid made from $\mathbf{v}_1, \mathbf{v}_2$, with corners $n\mathbf{v}_1 + m\mathbf{v}_2$ for integers n, m

Every point $\mathbf{x} \in \mathbf{R}^2$ lies inside some small blue parallelogram P, say the one with corners

$$n\mathbf{v}_1 + m\mathbf{v}_2, \ \ (n+1)\mathbf{v}_1 + m\mathbf{v}_2, \ \ n\mathbf{v}_1 + (m+1)\mathbf{v}_2, \ \ (n+1)\mathbf{v}_1 + (m+1)\mathbf{v}_2.$$

Some portion of the distance along each side-direction of P from the lower-left corner $n\mathbf{v}_1 + m\mathbf{v}_2$ pins down where $\mathbf{x}$ is located inside P: Figure 5.1.4 shows an $\mathbf{x}$ whose location relative to the lower-left corner $\mathbf{v}_1 + 2\mathbf{v}_2$ is $2/3$ of the distance along the $\mathbf{v}_1$-direction and $4/7$ of the distance along the $\mathbf{v}_2$-direction.

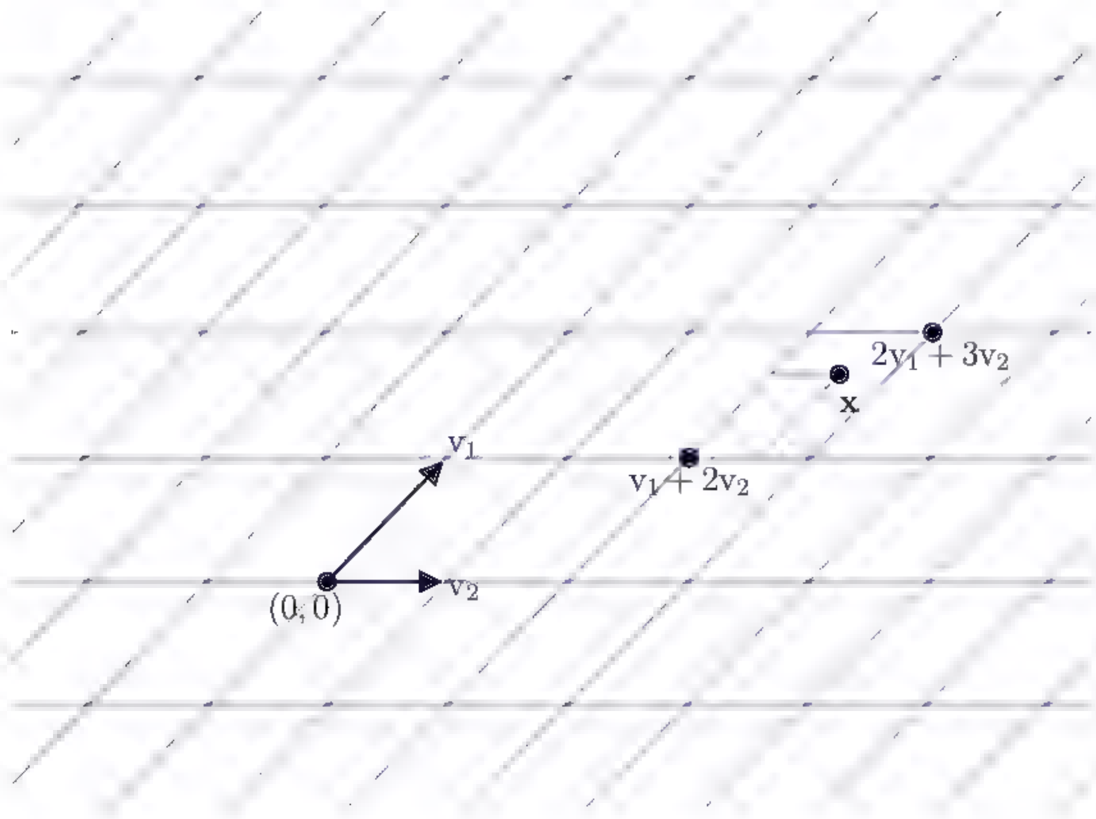


FIGURE 5.1.4. Point $\mathbf{x}$ inside a parallelogram some portion of the distance along each side direction from the corner $\mathbf{v}_1 + 2\mathbf{v}_2$: it is $2/3$ further along $\mathbf{v}_1$, $4/7$ further along $\mathbf{v}_2$

In such a case, we can write

$$\mathbf{x} = (\mathbf{v}_1 + 2\mathbf{v}_2) + ((2/3)\mathbf{v}_1 + (4/7)\mathbf{v}_2) = (5/3)\mathbf{v}_1 + (18/7)\mathbf{v}_2.$$

We have a similar geometric interpretation for any $a\mathbf{v}_1 + b\mathbf{v}_2$ by writing each of the scalars a and b as an "integer part" plus a "decimal part" (between 0 and 1). This explains the spanning property.

The smallest size of a spanning set for $\mathbf{R}^2$ is 2 since a single vector $\mathbf{v}$ certainly cannot span $\mathbf{R}^2$: such a span is a line (or a point if $\mathbf{v} = \mathbf{0}$). Hence, $\{\mathbf{v}_1, \mathbf{v}_2\}$ is a basis for $\mathbf{R}^2$. ■

In the cases $k = 1, 2, 3$ there is a reasonable way to figure out if the span of k nonzero vectors in $\mathbf{R}^n$ has dimension k (rather than $< k$). For $k = 1$ there is nothing to say (i.e., the span of a single nonzero vector is always 1-dimensional: there's nothing smaller the dimension could be), and for $k = 2, 3$ we use the following:

Dimension Criterion. For two nonzero vectors $\mathbf{v}, \mathbf{w} \in \mathbf{R}^n$ we have $\dim(\mathrm{span}(\mathbf{v}, \mathbf{w})) = 2$ except for precisely when the vectors are scalar multiples of each other, in which case the dimension is 1.

For three nonzero vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \in \mathbf{R}^n$ with span V, we have $\dim(V) = 3$ except for precisely the following cases:

(i) all three vectors are scalar multiples of each other, in which case $\dim(V) = 1$;
(ii) exactly two of the vectors are scalar multiples of each other, in which case $\dim(V) = 2$;
(iii) no $\mathbf{v}_i$ is a scalar multiple of another $\mathbf{v}_j$ but some $\mathbf{v}_i$ is a linear combination of the other two, in which case $\dim(V) = 2$ and *every* $\mathbf{v}_i$ is a linear combination of the other two.

Here are some examples to illustrate the preceding criterion (and a proof of the Dimension Criterion is given in Remark 5.1.9 for those who are interested).

Example 5.1.5. Consider the following pairs of nonzero vectors:

$$\mathbf{v} = \begin{bmatrix} 3/2 \\ -2 \end{bmatrix}, \mathbf{w} = \begin{bmatrix} -2 \\ 8/3 \end{bmatrix}; \quad \mathbf{v}' = \begin{bmatrix} 2 \\ -4 \\ 5 \end{bmatrix}, \mathbf{w}' = \begin{bmatrix} 3 \\ -6 \\ 15 \end{bmatrix}; \quad \mathbf{v}'' = \begin{bmatrix} 3 \\ 6 \\ -15 \end{bmatrix}, \mathbf{w}'' = \begin{bmatrix} 2 \\ 4 \\ -10 \end{bmatrix}.$$

To determine if the span of each pair (in $\mathbf{R}^2$ for the first pair, and in $\mathbf{R}^3$ for the second and third pairs) has dimension 1 or 2, we have to check in each case if the first vector is a scalar multiple of the second (or the other way around, whichever we prefer; the two options are equivalent since the vectors are all nonzero).

Does $\mathbf{v} = a\mathbf{w}$ for some scalar a? Let's write out what this means in terms of vector entries and try to figure out what a could be. The condition is

$$\begin{bmatrix} 3/2 \\ -2 \end{bmatrix} = a \begin{bmatrix} -2 \\ 8/3 \end{bmatrix} = \begin{bmatrix} -2a \\ 8a/3 \end{bmatrix},$$

or in other words the simultaneous conditions

$$3/2 = -2a, \quad -2 = 8a/3.$$

The first equation says $a = -3/4$, and that satisfies the second one ($8(-3/4)/3 = -8/4 = -2$), so $\text{span}(\mathbf{v}, \mathbf{w})$ is 1-dimensional.

For the next pair, the condition $\mathbf{v}' = a\mathbf{w}'$ for some unknown scalar a says

$$\begin{bmatrix} 2 \\ -4 \\ 5 \end{bmatrix} = \begin{bmatrix} 3a \\ -6a \\ 15a \end{bmatrix},$$

which amounts to *three* simultaneous conditions on a:

$$2 = 3a, \quad -4 = -6a, \quad 5 = 15a,$$

so we solve for a using one of these and check if it works for the others. The first equation says $a = 2/3$, and that satisfies the second equation ($-6(2/3) = -4$) but it fails the third equation: $15(2/3) = 10 \neq 5$. Hence, $\mathbf{v}'$ is *not* a scalar multiple of $\mathbf{w}'$, so $\text{span}(\mathbf{v}', \mathbf{w}')$ has dimension 2.

Finally, to check if $\mathbf{v}'' = a\mathbf{w}''$ for some scalar a we again write out the vector equation and equate corresponding entries, getting three simultaneous equations in a (one for each vector entry):

$$3 = 2a, \quad 6 = 4a, \quad -15 = -10a.$$

The first of these says $a = 3/2$, and this is readily checked to satisfy the other two equations (so $\mathbf{v}'' = (3/2)\mathbf{w}''$). Hence, $\text{span}(\mathbf{v}'', \mathbf{w}'')$ is 1-dimensional. ■

Example 5.1.6. Consider the triple of nonzero 5-vectors

$$\mathbf{v}_1 = \begin{bmatrix} -1 \\ 1 \\ -2/3 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 1 \\ 0 \\ -1/3 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} -2 \\ 0 \\ 1/3 \\ 0 \\ 1 \end{bmatrix}$$

from Example 4.1.13 that span W as defined there, so $\dim W \leq 3$. We shall now show that $\dim W = 3$. By the method of Example 5.1.5 one checks that no $\mathbf{v}_i$ is a scalar multiple of any $\mathbf{v}_j$ (it is easier than usual in this case because there are so many entries equal to 0; e.g., we can't have $\mathbf{v}_1 = a\mathbf{v}_2$ or $\mathbf{v}_1 = b\mathbf{v}_3$ due to comparing second entries, nor can we have $\mathbf{v}_2 = c\mathbf{v}_3$ due to comparing fourth entries).

Hence, we aren't in cases (i) or (ii) of the Dimension Criterion, so (by case (iii)) $\dim(W) < 3$ precisely when some $\mathbf{v}_i$ is a linear combination of the other two and then necessarily *every* $\mathbf{v}_i$ is a linear combination of the other two. Thus, to rule out this possibility there is no harm in ruling out just one of the possibilities:

$$\mathbf{v}_2 \overset{?}{=} a\mathbf{v}_1 + b\mathbf{v}_3$$

for some scalars a and b. We have to show there are no such scalars.

Equating corresponding entries in the vectors on both sides of such a hypothetical relationship gives five equations (one per vector entry) in 2 unknowns (a and b). We want to show that this system of many equations in few unknowns has no simultaneous solution. This is a variant on experience with 2 equations in 2 unknowns, as we shall see. The point is that we have a system of equations in only 2 unknowns, and this is something you learned how to manage in your prior study of algebra: solve for one variable in terms of the other using one of the equations, and plug that into the rest to see if they admit a common solution for that remaining variable.

Putting in the numbers, the vector equation of interest is

$$\begin{bmatrix} 1 \\ 0 \\ -1/3 \\ 1 \\ 0 \end{bmatrix} = a \begin{bmatrix} -1 \\ 1 \\ -2/3 \\ 0 \\ 0 \end{bmatrix} + b \begin{bmatrix} -2 \\ 0 \\ 1/3 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -a-2b \\ a \\ -2a/3+b/3 \\ 0 \\ b \end{bmatrix},$$

so comparing corresponding entries turns this into the system of simultaneous equations

$$1 = -a-2b, \quad 0 = a, \quad -1/3 = -2a/3 + b/3, \quad 1 = 0, \quad 0 = b.$$

There are many ways to see that this system has *no solution*. For instance, the fourth equation "$1 = 0$" is already impossible. Or alternatively the second and fifth equations tell us the values of a and b directly, but those don't work in the first or third equations (though it is enough to reach just one inconsistency, let alone multiple such, to conclude that there is no solution).

Since there is no solution, so $\mathbf{v}_2$ is not in the span of $\mathbf{v}_1$ and $\mathbf{v}_3$, we conclude that $\dim(W) = 3$. (If we had instead tried to write $\mathbf{v}_1 = a\mathbf{v}_2 + b\mathbf{v}_3$ for some scalars a, b then we would have arrived at a different system of 5 equations in 2 unknowns, but again we would have found that it has no simultaneous solution.) ■

Example 5.1.7. Consider the triple of nonzero 3-vectors

$$\mathbf{v}_1 = \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix} 0 \\ 2 \\ -1 \end{bmatrix}, \mathbf{v}_3 = \begin{bmatrix} 2 \\ -4 \\ 3 \end{bmatrix}$$

from Example 4.2.9. Does its span V have dimension equal to 3, or is $\dim(V) < 3$?

As in the preceding example, one first checks by inspection that no $\mathbf{v}_i$ is a scalar multiple of the others (please check this for yourself, arguing as in Example 5.1.5). Hence, again using the Dimension Criterion as in Example 5.1.6, we seek to determine if one of the $\mathbf{v}_i$'s is in the span of the other two. It doesn't matter which $\mathbf{v}_i$ we try, so we'll go with $\mathbf{v}_1$: is $\mathbf{v}_1 = a\mathbf{v}_2 + b\mathbf{v}_3$ for some scalars a and b? This says

$$\begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix} = a \begin{bmatrix} 0 \\ 2 \\ -1 \end{bmatrix} + b \begin{bmatrix} 2 \\ -4 \\ 3 \end{bmatrix} = \begin{bmatrix} 2b \\ 2a-4b \\ -a+3b \end{bmatrix},$$

which is a system of 3 equations (one per vector entry) in 2 unknowns (a and b):

$$3 = 2b, \quad -1 = 2a - 4b, \quad 2 = -a + 3b. \tag{5.1.1}$$

To handle this, we solve for b in terms of a (or a in terms of b) using one of the equations and plug that into the other two, seeing if we get a consistent solution. If so then $\mathbf{v}_1 \in \mathrm{span}(\mathbf{v}_2, \mathbf{v}_3)$ and $\dim(V) = 2$ by case (iii) of the Dimension Criterion. If not then $\dim(V) = 3$.

There are now several ways to proceed. The first equation in (5.1.1) says $b = 3/2$, and plugging this into the other two equations gives

$$-1 = 2a - 4(3/2) = 2a - 6, \;\; 2 = -a + 3(3/2) = -a + 9/2,$$

each of which has the same solution: $a = 5/2$. Hence, $\mathbf{v}_1 = (5/2)\mathbf{v}_2 + (3/2)\mathbf{v}_3$ (and $\dim(V) = 2$).

Alternatively, one could use the third equation in (5.1.1) to get $b = (1/3)(2 + a)$ and plug this into the other two to get two equations in a, which are each found to have the same common solution $a = 5/2$ (please check for yourself), and then we substitute that into $b = (1/3)(2 + a)$ to get $b = 3/2$ once again. Or we could have used the second equation in (5.1.1) rather than the third to get started, or we could have solved for a in terms of b instead, and so on. There are only two unknowns, so the algebra isn't bad no matter what we try to do. ■

In Chapter 19 we will give a way to remove redundancy from a spanning set for a linear subspace V of $\mathbf{R}^n$. Although we have a way to figure out the dimension of the span of 2 or 3 nonzero vectors, we have to confront the reality that for the span of 4 or more nonzero vectors in $\mathbf{R}^n$ it becomes rather cumbersome to figure out the dimension via algebra alone; we need another way. There is a particularly useful type of spanning set for any linear subspace V that is *always* guaranteed to be a basis of V, as we explain in the next section, and this will underlie a geometric method (in Chapter 19) to determine the dimension of the span of any number of vectors.

Remark 5.1.8 (online resource). The second video in "Essence of Linear Algebra" includes visualizations for the concept of "basis". (The end of that video introduces an additional concept called "linear (in)dependence" that we will discuss much later, in Chapter 19, so feel free to ignore that for now; it concerns the "redundancy" in spanning sets alluded to above.)

Remark 5.1.9 (optional). To prove the Dimension Criterion for the span of 2 or 3 nonzero vectors, first note that by definition the span V of k nonzero vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$ in $\mathbf{R}^n$ certainly has dimension at most k (as the dimension is the *smallest* size of a spanning set for V). By Theorem 4.2.5, we have $\dim(V) < k$ precisely when one of the $\mathbf{v}_i$'s can be dropped without affecting the span.

When $k = 2$, dropping some $\mathbf{v}_i$ without affecting the span means that V is the span of the remaining nonzero vector (so $\dim(V) = 1$): this means V is the span of either $\mathbf{v}_1$ or $\mathbf{v}_2$. But both $\mathbf{v}_i$'s belong to V, so if $V = \mathrm{span}(\mathbf{v}_1)$ then $\mathbf{v}_2 = a\mathbf{v}_1$ for some scalar a, and likewise if $V = \mathrm{span}(\mathbf{v}_2)$ then $\mathbf{v}_1 = b\mathbf{v}_2$ for some scalar b. In either case, the scalar is *nonzero* (since $\mathbf{v}_1, \mathbf{v}_2 \neq \mathbf{0}$), so that scalar can be moved to the other side as its reciprocal: $(1/a)\mathbf{v}_2 = \mathbf{v}_1$ or $(1/b)\mathbf{v}_1 = \mathbf{v}_2$ respectively.

Suppose $k = 3$. Since the $\mathbf{v}_i$'s are nonzero, if any is a scalar multiple of another then the scalar multiplier must be nonzero and so it can be moved to the other side as its reciprocal. Hence, in case (i) we have $\mathbf{v}_2 = a\mathbf{v}_1$ and $\mathbf{v}_3 = b\mathbf{v}_1$ for some scalars a, b. Then any linear combination of the $\mathbf{v}_i$'s can be written as $c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + c_3\mathbf{v}_3 = c_1\mathbf{v}_1 + c_2(a\mathbf{v}_1) + c_3(b\mathbf{v}_1) = (c_1 + ac_2 + bc_3)\mathbf{v}_1$, so $V = \mathrm{span}(\mathbf{v}_1)$ is 1-dimensional.

If we are in case (ii) then by relabeling we can assume it is $\mathbf{v}_2$ and $\mathbf{v}_3$ that are scalar multiples of each other, but not $\mathbf{v}_1$. Writing $\mathbf{v}_3 = a\mathbf{v}_2$, we have

$$c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + c_3\mathbf{v}_3 = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + c_3(a\mathbf{v}_2) = c_1\mathbf{v}_1 + (c_2 + ac_3)\mathbf{v}_2,$$

so $V = \mathrm{span}(\mathbf{v}_1, \mathbf{v}_2)$. But $\mathbf{v}_2$ is not a multiple of $\mathbf{v}_1$, so $\dim(V) = 2$ by the settled case $k = 2$.

Now suppose we aren't in cases (i) or (ii). By Theorem 4.2.5, we have $\dim(V) < 3$ precisely when some $\mathbf{v}_i$ can be dropped without affecting the span, yet all three vectors belong to the span by design, so this says that some $\mathbf{v}_i$ is in the span of the other two:

$$\mathbf{v}_i = a\mathbf{v}_{i'} + b\mathbf{v}_{i''}$$

for some scalars a, b and the other two indices i', i''. But those scalars a and b must *both* be nonzero, as otherwise $\mathbf{v}_i$ would be a multiple of one of the others (e.g., if $a = 0$ then $\mathbf{v}_i = b\mathbf{v}_{i''}$) and that can't happen since we're now away from cases (i) and (ii). Hence, by putting $\mathbf{v}_{i'}$ (resp. $\mathbf{v}_{i''}$) on one side of the equation, putting everything else on the other side, and then dividing by the nonzero coefficient of $\mathbf{v}_{i'}$ (resp. $\mathbf{v}_{i''}$), we see that each of $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ is in the span of the other two. This puts us in case (iii).

5.2. Orthogonal bases.

Definition 5.2.1. A collection of vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$ in $\mathbf{R}^n$ is called *orthogonal* if

$$\mathbf{v}_i \cdot \mathbf{v}_j = 0 \text{ whenever } i \neq j.$$

In words, the vectors are all perpendicular to one another.

Theorem 5.2.2. If $\mathbf{v}_1, \dots, \mathbf{v}_k$ is an orthogonal collection of **nonzero** vectors in $\mathbf{R}^n$ then it is a basis for $\mathrm{span}(\mathbf{v}_1, \dots, \mathbf{v}_k)$. In particular, $\mathrm{span}(\mathbf{v}_1, \dots, \mathbf{v}_k)$ then has dimension k and we call $\mathbf{v}_1, \dots, \mathbf{v}_k$ an *orthogonal basis* for its span. (A single nonzero vector is *always* an orthogonal basis for its span!)

The span of a collection of k vectors in $\mathbf{R}^n$ has dimension at most k (e.g., three vectors in $\mathbf{R}^3$ lying in a common plane through $\mathbf{0}$ have span with dimension less than 3). By Theorem 5.2.2, orthogonality is a useful way to guarantee that k given **nonzero** n-vectors have a k-dimensional span.

Theorem 5.2.2 is proved in Section B.1, for those who are interested.

Example 5.2.3. Consider the span V of the following three vectors in $\mathbf{R}^5$:

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 3 \\ 2 \\ 1 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 0 \\ 3 \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} 0 \\ 3 \\ 0 \\ 2 \\ 1 \end{bmatrix} \tag{5.2.1}$$

This collection of three vectors is not orthogonal, since, for example, $\mathbf{v}_1 \cdot \mathbf{v}_2 = 1 + 0 + 6 + 0 + 3 = 10$. The method based on the Dimension Criterion affirms that $\dim(V) = 3$, so the triple $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ is a basis of V, but we would like to give an *orthogonal* basis of V.

Consider the following three nonzero vectors in V (which admittedly we are now pulling out of thin air; the way they are found systematically will be addressed in Example 19.3.8):

$$\mathbf{w}_1 = \mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 3 \\ 2 \\ 1 \end{bmatrix}, \quad \mathbf{w}_2 = -2\mathbf{v}_1 + 3\mathbf{v}_2 = \begin{bmatrix} 1 \\ 3 \\ 0 \\ -4 \\ 7 \end{bmatrix}, \quad \mathbf{w}_3 = -9\mathbf{v}_1 - 24\mathbf{v}_2 + 75\mathbf{v}_3 = \begin{bmatrix} -33 \\ 201 \\ -75 \\ 132 \\ -6 \end{bmatrix}. \tag{5.2.2}$$

By direct computation all dot products $\mathbf{w}_i \cdot \mathbf{w}_j$ can be checked to vanish for $i \neq j$, so the nonzero vectors $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3$ form an orthogonal basis for their span.

Without already knowing that $\dim(V) = 3$, it can be directly checked that the span of the $\mathbf{w}_j$'s coincides with that of the $\mathbf{v}_i$'s as follows. Each $\mathbf{w}_j$ is a linear combination of the $\mathbf{v}_i$'s by design, and one can go in reverse to express the $\mathbf{v}_i$'s as linear combinations of the $\mathbf{w}_j$'s by systematically unraveling how the $\mathbf{w}_j$'s are defined in terms of the $\mathbf{v}_i$'s to arrive at:

$$\mathbf{v}_1 = \mathbf{w}_1, \quad \mathbf{v}_2 = \frac{1}{3}\mathbf{w}_2 + \frac{2}{3}\mathbf{w}_1, \quad \mathbf{v}_3 = \frac{1}{75}(\mathbf{w}_3 + 8\mathbf{w}_2 + 25\mathbf{w}_1).$$

(Don't worry about where these expressions came from: what matters is that one can really also express the $\mathbf{v}_i$'s as linear combinations of the $\mathbf{w}_j$'s, and we have handed you such expressions on a silver platter out of thin air for illustration purposes only. You won't be asked to reproduce such calculations, and we will revisit this matter in a systematic way in Section 19.3.)

The upshot is that the linear subspace $V = \mathrm{span}(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ has $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ as an orthogonal spanning set, and by Theorem 5.2.2 an orthogonal set of nonzero vectors in $\mathbf{R}^n$ is always a basis of its span. Hence, the $\mathbf{w}_j$'s constitute a basis for V, giving another verification that $\dim(V) = 3$. The merit of this approach over the one based on the Dimension Criterion is that it will work well when the number of $\mathbf{v}_i$'s is much larger than 3, whereas the Dimension Criterion is only applicable to spans of at most three vectors. ■

There is a systematic process for finding an orthogonal basis for the span of k vectors in $\mathbf{R}^n$ called the "Gram[8]–Schmidt[9] process". (When applied to $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ in Example 5.2.3 it produces $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ in (5.2.2) up to nonzero scaling factors on each $\mathbf{w}_j$.) We will discuss this process in Chapter 19.

Example 5.2.4. Let's revisit the span V of the vectors

$$\mathbf{v}_1 = \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 0 \\ 2 \\ -1 \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} 2 \\ -4 \\ 3 \end{bmatrix}$$

considered in Example 4.2.9, where we essentially asked if $\dim(V) = 3$ or perhaps $\dim(V) < 3$. In Example 5.1.7 we used the Dimension Criterion to determine that $\dim(V) = 2$, finding along the way an explicit description of some $\mathbf{v}_i$ as a linear combination of the other two. Let's exhibit an orthogonal basis for V, giving another way to see that $\dim(V) = 2$ (the point being that this method will adapt to spans of any number of vectors, whereas the Dimension Criterion only applies to spans of at most three vectors).

We'll try to make an orthogonal basis of V consisting of vectors

$$\mathbf{w}_1 = \mathbf{v}_1 = \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix}, \quad \mathbf{w}_2 = \mathbf{v}_2 + c\mathbf{v}_1$$

for some scalar c. (This will be done systematically in Section 7.1, but here we just treat this specific example directly.) The orthogonality condition says $0 = \mathbf{w}_1 \cdot \mathbf{w}_2 = \mathbf{v}_1 \cdot (\mathbf{v}_2 + c\mathbf{v}_1) = \mathbf{v}_1 \cdot \mathbf{v}_2 + c(\mathbf{v}_1 \cdot \mathbf{v}_1)$. From the definitions of $\mathbf{v}_1$ and $\mathbf{v}_2$ we get $\mathbf{v}_1 \cdot \mathbf{v}_1 = 14$ and $\mathbf{v}_1 \cdot \mathbf{v}_2 = -4$, so $0 = -4 + c(14)$, which says $c = 2/7$. Hence,

$$\mathbf{w}_2 = \mathbf{v}_2 + \frac{2}{7}\mathbf{v}_1 = \begin{bmatrix} 0 \\ 2 \\ -1 \end{bmatrix} + \frac{2}{7}\begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 6/7 \\ 12/7 \\ -3/7 \end{bmatrix}.$$

Since $\mathbf{w}_1, \mathbf{w}_2$ belong to V, everything in the plane $\mathcal{P} = \mathrm{span}(\mathbf{w}_1, \mathbf{w}_2)$ belongs to V. Let's exhibit explictly why this span exhausts V (we know it must since $\dim V = 2$, but we want to explain the exhaustion in more direct terms just for the sake of being explicit – in Chapter 19 we will make systematic what we are about to do in a specific case). This amounts to showing that each of the $\mathbf{v}_j$'s belongs to $\mathcal{P}$, since then every linear combination of the $\mathbf{v}_j$'s would belong to $\mathcal{P}$, which is to say everything in V would belong to $\mathcal{P}$ as desired. By design we have $\mathbf{v}_1 = \mathbf{w}_1$, and by definition $\mathbf{w}_2 = \mathbf{v}_2+(2/7)\mathbf{v}_1 = \mathbf{v}_2+(2/7)\mathbf{w}_1$, so $\mathbf{v}_2 = \mathbf{w}_2 - (2/7)\mathbf{w}_1$. But for what scalars a_1, a_2 do we have $\mathbf{v}_3 = a_1\mathbf{w}_1 + a_2\mathbf{w}_2$?

---

[8] Jorgen Gram (1850-1916) was a Danish mathematician who worked in applied math, probability, and number theory. He earned no degree past a Masters and held no university position; he did mathematical research as an educated amateur in his spare time!

[9] Erhard Schmidt (1876-1959) was a German mathematician who worked on differential equations and related topics.

Here is a neat idea, which we will turn into a general method in Theorem 5.3.6: to exploit the orthogonality of $\mathbf{w}_1$ and $\mathbf{w}_2$ let's form the dot product of $\mathbf{v}_3 = a_1\mathbf{w}_1 + a_2\mathbf{w}_2$ against $\mathbf{w}_1$ and against $\mathbf{w}_2$:

$$\mathbf{v}_3 \cdot \mathbf{w}_1 = (a_1\mathbf{w}_1 + a_2\mathbf{w}_2) \cdot \mathbf{w}_1 = a_1(\mathbf{w}_1 \cdot \mathbf{w}_1) + a_2(\mathbf{w}_2 \cdot \mathbf{w}_1) = a_1(\mathbf{w}_1 \cdot \mathbf{w}_1) + a_2(0) = a_1(\mathbf{w}_1 \cdot \mathbf{w}_1),$$

so a_2 has *disappeared*. Plugging in the values $\mathbf{v}_3 \cdot \mathbf{w}_1 = 16$ and $\mathbf{w}_1 \cdot \mathbf{w}_1 = 14$ gives $16 = 14a_1$, so $a_1 = 16/14 = 8/7$. Similar dot product work yields $\mathbf{v}_3 \cdot \mathbf{w}_2 = a_2(\mathbf{w}_2 \cdot \mathbf{w}_2)$ with $\mathbf{v}_3 \cdot \mathbf{w}_2 = -45/7$ and $\mathbf{w}_2 \cdot \mathbf{w}_2 = 189/49 = 27/7$. Thus, $-45/7 = a_2(27/7)$, so $a_2 = -45/27 = -5/3$. In other words, $\mathbf{v}_3 = (8/7)\mathbf{w}_1 - (5/3)\mathbf{w}_2$.

The fact that the span $V = \text{span}(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ is not 3-dimensional can be expressed in a couple of other ways:

(i) First of all, you can verify the equality

$$\mathbf{v}_3 = (2/3)\mathbf{v}_1 - (5/3)\mathbf{v}_2$$

(which can be discovered by the method in Example 5.1.7, but that doesn't matter when you have been given the equality to be checked). We can use this expression for $\mathbf{v}_3$ in terms of $\mathbf{v}_1$ and $\mathbf{v}_2$ to re-express any linear combination of the $\mathbf{v}_i$'s in terms of $\mathbf{v}_1$ and $\mathbf{v}_2$, as follows:

$$a\mathbf{v}_1 + b\mathbf{v}_2 + c\mathbf{v}_3 = a\mathbf{v}_1 + b\mathbf{v}_2 + c((2/3)\mathbf{v}_1 - (5/3)\mathbf{v}_2) = (a + 2c/3)\mathbf{v}_1 + (b - 5c/3)\mathbf{v}_2, \qquad (5.2.3)$$

which is a span of just two vectors (namely, $\mathbf{v}_1$ and $\mathbf{v}_2$), so $\dim(V) \leq 2$. This expresses that the triple $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ as a spanning set for V is "inefficient" or "redundant" in the sense that if we drop $\mathbf{v}_3$ then the span of what remains is unaffected (it is still V).

(ii) Alternatively, one can verify directly that all of the $\mathbf{v}_i$'s belong to the plane through $\mathbf{0}$ given by

$$-x + y + 2z = 0.$$

(Don't worry about how one would find that equation, but with the equation in hand please check that the $\mathbf{v}_i$'s indeed lie in that plane.) Hence, any linear combination of the $\mathbf{v}_i$'s belongs to the same plane (by visualizing addition in terms of the parallelogram law), so V lies inside that plane.

■

The ubiquity of orthogonal bases is encoded in the following general fact:

Theorem 5.2.5. Every nonzero linear subspace of $\mathbf{R}^n$ has an orthogonal basis.

There is an especially convenient type of orthogonal basis for a nonzero linear subspace of $\mathbf{R}^n$:

Definition 5.2.6. A collection of vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ in $\mathbf{R}^n$ is called *orthonormal* if they are orthogonal to each other and *in addition* they are all unit vectors; that is, $\mathbf{v}_i \cdot \mathbf{v}_i = 1$ for all i (ensuring $\|\mathbf{v}_i\| = \sqrt{\mathbf{v}_i \cdot \mathbf{v}_i} = \sqrt{1} = 1$ for all i).

Any orthonormal collection of vectors is a basis of its span, by Theorem 5.2.2.

We will construct orthogonal (and orthonormal) bases from spanning sets, thereby establishing Theorem 5.2.5, in Chapter 19 (this is the Gram–Schmidt process to which we have alluded several times).

5.3. Examples of orthogonal bases. Here are examples illustrating the notion of orthonormal basis.

Example 5.3.1. The collection $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\2\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\3\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\4\end{bmatrix}$ is an orthogonal set of 4 vectors in $\mathbf{R}^4$, and it spans the entirety of $\mathbf{R}^4$ since we can write

$$\begin{bmatrix}a\\b\\c\\d\end{bmatrix} = a\begin{bmatrix}1\\0\\0\\0\end{bmatrix} + \frac{b}{2}\begin{bmatrix}0\\2\\0\\0\end{bmatrix} + \frac{c}{3}\begin{bmatrix}0\\0\\3\\0\end{bmatrix} + \frac{d}{4}\begin{bmatrix}0\\0\\0\\4\end{bmatrix}$$

for any $a, b, c, d \in \mathbf{R}$ (combine terms on the right side to see that indeed this always holds). Hence, $\mathbf{R}^4$ has dimension 4 (as we expect).

This orthogonal basis of $\mathbf{R}^4$ is not orthonormal because the vectors in it don't all have length 1. To obtain an orthonormal basis, we can divide each of these vectors by its length to arrive at an orthogonal basis of unit vectors:

$$\mathbf{e}_1 = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \mathbf{e}_3 = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \mathbf{e}_4 = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}. \tag{5.3.1}$$

■

Example 5.3.2. For any n the analogous orthonormal collection of n vectors $\mathbf{e}_1, \dots, \mathbf{e}_n$ in $\mathbf{R}^n$ can be written down (i.e., $\mathbf{e}_i$ has its ith entry equal to 1 and all other entries are 0), and this spans $\mathbf{R}^n$ much as for the case $n = 4$ above; it is called the *standard basis* of $\mathbf{R}^n$, and shows $\dim \mathbf{R}^n = n$ (as we expect). In particular, by Theorem 4.2.8 (with $V = \mathbf{R}^n$), every linear subspace of $\mathbf{R}^n$ has dimension at most n and the only n-dimensional one is $\mathbf{R}^n$ itself (as geometric intuition may suggest).

In the special case $n = 3$, the vectors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \in \mathbf{R}^3$ are often respectively denoted as $\mathbf{i}, \mathbf{j}, \mathbf{k}$ in physics and engineering contexts. ■

There are *lots* of orthogonal bases for $\mathbf{R}^n$ other than the basis $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$ introduced in Example 5.3.2. Here are examples for some values of n.

Example 5.3.3. The triple $\begin{bmatrix}1\\2\\4\end{bmatrix}, \begin{bmatrix}-6\\1\\1\end{bmatrix}, \begin{bmatrix}-2\\-25\\13\end{bmatrix}$ is an orthogonal basis for $\mathbf{R}^3$. How can one "see" this? One doesn't check by hand that its span coincides with $\mathbf{R}^3$ (that would get bogged down in a messy system of 3 equations in 3 unknowns.). Rather, one can check by hand that it is an orthogonal collection of vectors (please check), so this collection of 3 nonzero vectors must be a basis of its span by Theorem 5.2.2, and hence its span has dimension 3. But we have noted in Example 5.3.2 that for any n the only n-dimensional subspace of $\mathbf{R}^n$ is itself, so the 3-dimensional span inside $\mathbf{R}^3$ is indeed $\mathbf{R}^3$. Observe the power of vector algebra!

This triple is not orthonormal because the vectors don't have length 1. We get an orthonormal basis by scaling them to be unit vectors (dividing each by its length):

$$\frac{1}{\sqrt{21}}\begin{bmatrix}1\\2\\4\end{bmatrix}, \frac{1}{\sqrt{38}}\begin{bmatrix}-6\\1\\1\end{bmatrix}, \frac{1}{\sqrt{798}}\begin{bmatrix}-2\\-25\\13\end{bmatrix}. \tag{5.3.2}$$

The procedure of dividing by the length to get unit vectors typically leads to messy expressions with square roots. That is one reason that whenever possible we prefer to work with orthogonal bases rather than further demanding orthonormality (i.e., the unit length condition). ■

Example 5.3.4. The quadruple of nonzero vectors

$$\begin{bmatrix}-1\\1\\1\\-1\end{bmatrix}, \begin{bmatrix}0\\0\\1\\1\end{bmatrix}, \begin{bmatrix}1\\1\\0\\0\end{bmatrix}, \begin{bmatrix}1\\-1\\1\\-1\end{bmatrix}$$

is an orthogonal basis for $\mathbf{R}^4$: one can check orthogonality by computing the dot products, and then argue just as in Example 5.3.3. This basis is not orthonormal because the vectors don't have length 1. We get an orthonormal basis by scaling them to be unit vectors:

$$\frac{1}{2}\begin{bmatrix}-1\\1\\1\\-1\end{bmatrix}, \frac{1}{\sqrt{2}}\begin{bmatrix}0\\0\\1\\1\end{bmatrix}, \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\\0\\0\end{bmatrix}, \frac{1}{2}\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix}.$$

■

Example 5.3.5. If $\mathbf{v}_1, \dots, \mathbf{v}_k$ are nonzero vectors in $\mathbf{R}^n$, by definition any vector $\mathbf{v} \in \mathrm{span}(\mathbf{v}_1, \dots, \mathbf{v}_k)$ can be written as a linear combination

$$\mathbf{v} = \sum_{i=1}^{k} c_i \mathbf{v}_i \tag{5.3.3}$$

for some scalars $c_1, \dots, c_k$. If the collection of $\mathbf{v}_i$'s is *orthogonal*, we can actually *solve for the c_i's in terms of* $\mathbf{v}$ by the following slick technique that has useful generalizations throughout mathematics (with Fourier series, special function theory, and so on). The method is to form the dot product of both sides of equation (5.3.3) against each $\mathbf{v}_i$ separately.

For instance, if we form the dot product against $\mathbf{v}_1$ then we obtain

$$\mathbf{v} \cdot \mathbf{v}_1 = c_1(\mathbf{v}_1 \cdot \mathbf{v}_1) + c_2 \underbrace{(\mathbf{v}_2 \cdot \mathbf{v}_1)}_{=0} + c_3 \underbrace{(\mathbf{v}_3 \cdot \mathbf{v}_1)}_{=0} + \cdots = c_1(\mathbf{v}_1 \cdot \mathbf{v}_1),$$

where the tremendous cancellation at the final equality is precisely due to the *orthogonality* of the collection of $\mathbf{v}_i$'s. Since $\mathbf{v}_1$ is nonzero, so $\mathbf{v}_1 \cdot \mathbf{v}_1 = \|\mathbf{v}_1\|^2$ is nonzero, we can now divide by it at both ends of our string of equalities above to obtain

$$\frac{\mathbf{v} \cdot \mathbf{v}_1}{\mathbf{v}_1 \cdot \mathbf{v}_1} = c_1.$$

In this way we have solved for c_1!

The same procedure works likewise to solve for each c_i via forming dot products against $\mathbf{v}_i$, yielding the general formula

$$c_i = \frac{\mathbf{v} \cdot \mathbf{v}_i}{\mathbf{v}_i \cdot \mathbf{v}_i} \tag{5.3.4}$$

for each i. Substituting back into the right side of (5.3.3), we obtain the following result.

Theorem 5.3.6 (Fourier[10] formula). For any orthogonal collection of nonzero vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$ in $\mathbf{R}^n$ and vector $\mathbf{v}$ in their span,

$$\mathbf{v} = \sum_{i=1}^{k} \left(\frac{\mathbf{v} \cdot \mathbf{v}_i}{\mathbf{v}_i \cdot \mathbf{v}_i} \right) \mathbf{v}_i. \tag{5.3.5}$$

In particular, if the $\mathbf{v}_i$'s are all unit vectors (so $\mathbf{v}_i \cdot \mathbf{v}_i = 1$ for all i) then $\mathbf{v} = \sum_{i=1}^{k} (\mathbf{v} \cdot \mathbf{v}_i)\mathbf{v}_i$.

WARNING: Make sure you understand the notation in (5.3.5), since it is mixing scalars – the ratio of dot products – and vectors. Example 5.3.8 gives an interesting numerical illustration.

■

Let's illustrate the Fourier formula (5.3.5) in some examples.

Example 5.3.7. For the orthonormal basis $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4$ of $\mathbf{R}^4$ as in (5.3.1) and any $\mathbf{v} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \in \mathbf{R}^4$, the coefficients $\mathbf{v} \cdot \mathbf{e}_i$ in (5.3.5) work out as follows:

$$\mathbf{v} \cdot \mathbf{e}_1 = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = a_1$$

and similarly $\mathbf{v} \cdot \mathbf{e}_i = a_i$ for each $i = 1, 2, 3, 4$. Thus (since $\mathbf{e}_i \cdot \mathbf{e}_i = 1$), $\mathbf{v} = \sum_{i=1}^{4} (\mathbf{v} \cdot \mathbf{e}_i)\mathbf{e}_i = \sum_{i=1}^{4} a_i \mathbf{e}_i$.

Unpacking the summation notation, this is just asserting

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = a_1 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + a_2 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + a_3 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + a_4 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

which can be directly verified by hand since the right side is exactly

$$\begin{bmatrix} a_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ a_2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ a_3 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ a_4 \end{bmatrix}.$$

In other words, the Fourier formula (5.3.5) in the special case that $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ is the orthonormal basis $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ of $\mathbf{R}^n$ is precisely the familiar fact that any vector in $\mathbf{R}^n$ can be decomposed as the sum of its "components" along the various coordinate directions. This is neither surprising nor perhaps particularly interesting, so we next give a more "typical" example. ■

Example 5.3.8. Consider the explicit orthogonal basis $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ given in (5.2.2) for the linear subspace $V = \mathrm{span}(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ in $\mathbf{R}^5$ with explicit $\mathbf{v}_i$'s as defined at the start of Example 5.2.3. Applying the Fourier formula in (5.3.5) for the orthogonal basis $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ of V, for each $\mathbf{v} \in V$ we have

$$\mathbf{v} = \sum_{i=1}^{3} \left(\frac{\mathbf{v} \cdot \mathbf{w}_i}{\mathbf{w}_i \cdot \mathbf{w}_i} \right) \mathbf{w}_i = \frac{\mathbf{v} \cdot \mathbf{w}_1}{15}\mathbf{w}_1 + \frac{\mathbf{v} \cdot \mathbf{w}_2}{75}\mathbf{w}_2 + \frac{\mathbf{v} \cdot \mathbf{w}_3}{64575}\mathbf{w}_3 \tag{5.3.6}$$

since the denominators in the final expression are just the dot product evaluations

$$\mathbf{w}_1 \cdot \mathbf{w}_1 = 15, \quad \mathbf{w}_2 \cdot \mathbf{w}_2 = 75, \quad \mathbf{w}_3 \cdot \mathbf{w}_3 = 64575.$$

---

[10] Joseph Fourier (1768-1830) was a French mathematical physicist who worked on differential equations, function theory, and the theory of heat. His bold claim that every periodic function is an infinite series in sines and cosines (a "Fourier series") revolutionized math and physics; the ratio of dot products in (5.3.5) is deeply related to a formula for coefficients of Fourier series. In 1802 he showed a copy of the Rosetta Stone to an 11-year-old J-F. Champollion, who deciphered it 20 years later.

To illustrate the usefulness of (5.3.6), consider the vector

$$\mathbf{v} = 2\mathbf{v}_1 - \mathbf{v}_2 + \mathbf{v}_3 = \begin{bmatrix} 2 \\ 0 \\ 6 \\ 4 \\ 2 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 2 \\ 0 \\ 3 \end{bmatrix} + \begin{bmatrix} 0 \\ 3 \\ 0 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 6 \\ 0 \end{bmatrix}$$

in V. Since $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ is a basis of V, we know that there is some expression of the form

$$\mathbf{v} = c_1\mathbf{w}_1 + c_2\mathbf{w}_2 + c_3\mathbf{w}_3$$

for unknown scalars c_1, c_2, c_3. What are these scalars? A brute-force approach would be to write everything out as explicit vectors to obtain

$$\begin{bmatrix} 1 \\ 2 \\ 4 \\ 6 \\ 0 \end{bmatrix} = \mathbf{v} = c_1\mathbf{w}_1 + c_2\mathbf{w}_2 + c_3\mathbf{w}_3 = c_1 \begin{bmatrix} 1 \\ 0 \\ 3 \\ 2 \\ 1 \end{bmatrix} + c_2 \begin{bmatrix} 1 \\ 3 \\ 0 \\ -4 \\ 7 \end{bmatrix} + c_3 \begin{bmatrix} -33 \\ 201 \\ -75 \\ 132 \\ -6 \end{bmatrix} = \begin{bmatrix} c_1 + c_2 - 33c_3 \\ 3c_2 + 201c_3 \\ 3c_1 - 75c_3 \\ 2c_1 - 4c_2 + 132c_3 \\ c_1 + 7c_2 - 6c_3 \end{bmatrix},$$

and then equate corresponding vector entries on the left and right sides to get a huge system of 5 equations in 3 unknowns. We can *entirely bypass* that by computing dot products as in (5.3.6) for our specific $\mathbf{v}$!

To carry this out, we use the explicit descriptions of $\mathbf{v}$ and the $\mathbf{w}_i$'s to compute

$$\mathbf{v} \cdot \mathbf{w}_1 = 25, \quad \mathbf{v} \cdot \mathbf{w}_2 = -17, \quad \mathbf{v} \cdot \mathbf{w}_3 = 861,$$

so (5.3.6) says for this particular $\mathbf{v}$ that

$$\mathbf{v} = \frac{25}{15}\mathbf{w}_1 - \frac{17}{75}\mathbf{w}_2 + \frac{861}{64575}\mathbf{w}_3 = \frac{5}{3}\mathbf{w}_1 - \frac{17}{75}\mathbf{w}_2 + \frac{1}{75}\mathbf{w}_3. \tag{5.3.7}$$

That's it! This is the expression for $\mathbf{v}$ as a linear combination of the orthogonal basis $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ of V. (Don't worry: we won't ask you to carry out a calculation involving such big numbers and complicated fractions by hand, and in particular nothing like this on an exam.)

As a safety check, or to impress ourselves with the consistency of mathematics, the equality (5.3.7) can be verified by direct numerical calculation of the right side using the definition of the $\mathbf{w}_j$'s in (5.2.2):

$$\frac{5}{3}\mathbf{w}_1 - \frac{17}{75}\mathbf{w}_2 + \frac{1}{75}\mathbf{w}_3 = \frac{5}{3}\mathbf{w}_1 + \frac{1}{75}(-17\mathbf{w}_2 + \mathbf{w}_3) = \frac{5}{3}\mathbf{w}_1 + \frac{1}{75}\left(-17 \begin{bmatrix} 1 \\ 3 \\ 0 \\ -4 \\ 7 \end{bmatrix} + \begin{bmatrix} -33 \\ 201 \\ -75 \\ 132 \\ -6 \end{bmatrix} \right)$$

$$= \frac{5}{3}\mathbf{w}_1 + \frac{1}{75} \begin{bmatrix} -50 \\ 150 \\ -75 \\ 200 \\ -125 \end{bmatrix}.$$

All entries in the column vector are divisible by 25, which is also a factor of 75, so cancelling 25's turns this into

$$\frac{5}{3}\mathbf{w}_1 + \frac{1}{3}\begin{bmatrix} -2 \\ 6 \\ -3 \\ 8 \\ -5 \end{bmatrix} = \frac{1}{3}\left(5\mathbf{w}_1 + \begin{bmatrix} -2 \\ 6 \\ -3 \\ 8 \\ -5 \end{bmatrix}\right) = \frac{1}{3}\left(5\begin{bmatrix} 1 \\ 0 \\ 3 \\ 2 \\ 1 \end{bmatrix} + \begin{bmatrix} -2 \\ 6 \\ -3 \\ 8 \\ -5 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 6 \\ 0 \end{bmatrix},$$

which is exactly **v** as claimed. ■

Chapter 5 highlights (links to highlights in and next chapters)

Notation	**Meaning**	**Location in text**
$\mathbf{e}_j$ as an n-vector	the n-vector with entries all 0 except for 1 in jth entry	(5.3.1), Example 5.3.2

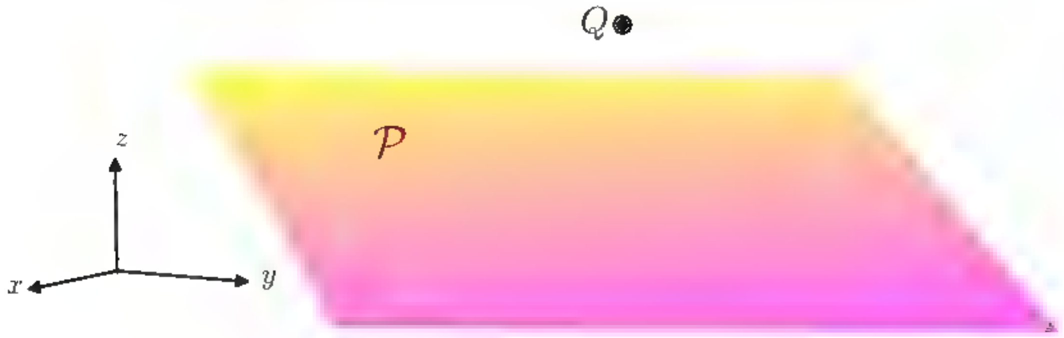
Concept	**Meaning**	**Location in text**
basis for a nonzero linear subspace V of $\mathbf{R}^n$	a spanning set of least possible size for V	Definition 5.1.1
standard basis of $\mathbf{R}^n$	the collection of n-vectors $\mathbf{e}_1, \dots, \mathbf{e}_n$	Definition 5.3.2
orthogonal collection of n-vectors	a collection of n-vectors that are each orthogonal to the rest	Definition 5.2.1
orthogonal basis (for a nonzero linear subspace V of $\mathbf{R}^n$)	a basis of V consisting of pairwise orthogonal n-vectors	Theorem 5.2.2
orthonormal collection of n-vectors	unit n-vectors that are pairwise orthogonal to each other	Definition 5.2.6

Result	**Meaning**	**Location in text**
Dimension Criterion	characterizes possibilities for the dimension of the span of two or three nonzero n-vectors	box just above Example 5.1.5
any orthogonal collection of **nonzero** n-vectors is basis of its span	if $\mathbf{v}_1, \dots, \mathbf{v}_k \in \mathbf{R}^n$ are nonzero and pairwise orthogonal then $\mathrm{span}(\mathbf{v}_1, \dots, \mathbf{v}_k)$ has dimension k	Theorem 5.2.2
orthogonal basis always exists	every nonzero linear subspace V in $\mathbf{R}^n$ has a basis consisting of pairwise orthogonal vectors	Theorem 5.2.5
$\dim \mathbf{R}^n = n$	the least size of a spanning set for $\mathbf{R}^n$ is n (and the only n-dimensional linear subspace of $\mathbf{R}^n$ is itself!)	Example 5.3.2
Fourier formula	for an orthogonal collection $\mathbf{v}_1, \dots, \mathbf{v}_k$ of nonzero n-vectors and any $\mathbf{v}$ in their span, explicit formula for scalars $c_1, \dots, c_k$ so that $\mathbf{v} = c_1\mathbf{v}_1 + \cdots + c_k\mathbf{v}_k$	Theorem 5.3.6

Skill	**Location in text**
visualize basis of $\mathbf{R}^2$ or $\mathbf{R}^3$ as a gridline description	Figure 5.1.1 (for $\mathbf{R}^3$), Example 5.1.4 (for $\mathbf{R}^2$)
use algebra to apply Dimension Criterion	Examples 5.1.5–5.1.7
verify if n given n-vectors constitute an orthogonal (or orthonormal) basis of $\mathbf{R}^n$	Examples 5.3.1–5.3.4

6. Projections

In this chapter we study questions of the following type: for a point Q and plane $\mathcal{P}$ in $\mathbf{R}^3$, what point on $\mathcal{P}$ is closest to Q? (This general problem can be reduced to the case when $\mathcal{P}$ passes through $\mathbf{0}$.)



For many applications in engineering, physics, and data-related problems in all scientific fields (genetics, economics, neuroscience, computer science, etc.) it is *essential* to go far beyond $\mathbf{R}^3$ and solve similar problems involving distance minimization to subspaces in $\mathbf{R}^n$ for any n:

> If V is a linear subspace in $\mathbf{R}^n$ and $\mathbf{x} \in \mathbf{R}^n$ is some point, then what point in V is closest to $\mathbf{x}$? (This closest point will be called the *projection* of $\mathbf{x}$ into V.)

We saw instances of the utility of the answer to this in economics and neuroscience in Example 4.1.9 (the application to facial recognition with $n = 50$ given in [CT] is discussed further in Example 6.2.2). In Chapter 7 we will use it to find the line that best fits n data points in $\mathbf{R}^2$ (solving a practical problem in $\mathbf{R}^2$ using geometric ideas in $\mathbf{R}^n$!), building on work in this chapter using distance minimization to make orthogonal bases in special cases. The pervasive data-analysis technique called *principal component analysis* (PCA) to be discussed in Section 27.3 is another important real-world application of the methods developed in this chapter.

By the end of this chapter, you should be able to:

- compute the point on a line in $\mathbf{R}^n$ through $\mathbf{0}$ that is closest to a given point in $\mathbf{R}^n$;
- compute projection into a subspace V of $\mathbf{R}^n$ when given an orthogonal basis for the subspace.

6.1. The closest point to a line. The way we will solve the general problem of distance minimization from a point in $\mathbf{R}^n$ to a linear subspace V involves "assembling" a collection of solutions to distance minimization to certain 1-dimensional linear subspaces of V, or in other words, solving distance minimization problems to a collection of lines through $\mathbf{0}$ in $\mathbf{R}^n$.

Since minimizing distance to lines will be the foundation for the general case, we begin by focusing on this special case. Consider a line L in $\mathbf{R}^n$ through $\mathbf{0}$, so $L = \mathrm{span}(\mathbf{w}) = \{c\mathbf{w} : c \in \mathbf{R}\}$ where $\mathbf{w} \in \mathbf{R}^n$ is some nonzero vector. For any point $\mathbf{x} \in \mathbf{R}^n$, we want to show that there is a unique point in L closest to $\mathbf{x}$, and to actually give a formula for how to compute this nearest point to $\mathbf{x}$ in L. How can we get an idea about what to do? That is:

(i) Why should we believe that in $\mathbf{R}^n$ for general n there is a (unique) point on L closest to $\mathbf{x}$?
(ii) How can we determine what that point is? (It will be denoted $\mathbf{Proj}_L(\mathbf{x})$ or $\mathbf{Proj}_{\mathbf{w}}(\mathbf{x})$, and called the *projection* of $\mathbf{x}$ into L or into $\mathbf{w}$.)

Here is the fundamental idea: although our task (necessary for many applications!) takes place in $\mathbf{R}^n$ with completely general (and possibly huge) n, we look at a *low-dimensional* instance of the problem in the hope that the low-dimensional case will suggest some feature that has a chance to adapt to the general situation. This balancing of insight from pictures in low-dimensional cases alongside algebraic work and geometric language developed for $\mathbf{R}^n$ with general n is an important part of linear algebra, giving visual insight into $\mathbf{R}^n$ for big n. (This doesn't justify results in $\mathbf{R}^n$ for general n, but it inspires what we should expect and/or try to prove is true.)

To this end, let's look at the case $n = 2$ shown in Figure 6.1.1 below.

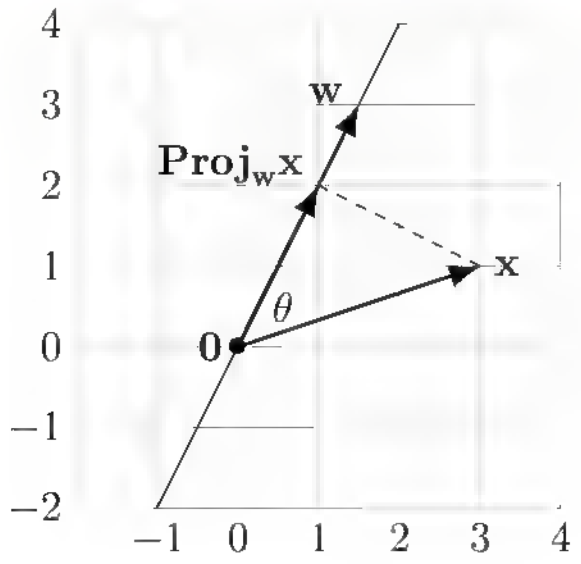


FIGURE 6.1.1. Projection of the vector $\mathbf{x}$ onto the line L spanned by $\mathbf{w}$

The key insight suggested by Figure 6.1.1 is that the point on the line L that is closest to $\mathbf{x}$ has *another* characterization (that in turn will allow us to compute it): it is the one point on L for which the displacement vector to $\mathbf{x}$ (the dotted line segment joining it to $\mathbf{x}$ as in Figure 6.1.1) is *perpendicular* to L, or equivalently is perpendicular to $\mathbf{w}$. Visually, the perpendicular direction to L from $\mathbf{x}$ is the "most direct" route. Or put another way, you may convince yourself by drawing some pictures that any deviation from perpendicularity entails a longer path from $\mathbf{x}$ to the line L.

To summarize, Figure 6.1.1 suggests a workable idea: *the point $c\mathbf{w} \in L$ for which $\|\mathbf{x} - c\mathbf{w}\|$ is minimal should also have the property that $\mathbf{x} - c\mathbf{w}$ is orthogonal to everything in L.* Although this idea is suggested by the picture in $\mathbf{R}^2$, as written it makes equally good sense in $\mathbf{R}^n$ for any n whatsoever. But is it true? (Certainly a picture in $\mathbf{R}^2$, no matter how convincing, does not justify that this is still valid in $\mathbf{R}^n$ for general n.) And even once we know it is true, how can we exploit this property of the nearest point to $\mathbf{x}$ on L to actually *compute* this nearest point?

The informal reasoning above may have already convinced you that the distance is minimized precisely when the displacement vector is orthogonal to L. We now explain two ways to use that characterization to obtain a clean and explicit formula for the nearest point. The results of these are then summarized in Proposition 6.1.1, and we follow with some worked examples.

Method I (algebraic). In accordance with the idea inspired by Figure 6.1.1 in the case $n = 2$, for general n we look for a scalar c for which $\mathbf{x} - c\mathbf{w}$ is orthogonal to every vector in L. The points of $L = \operatorname{span}(\mathbf{w})$ are those of the form $a\mathbf{w}$ for scalars a, so we seek c making $(\mathbf{x} - c\mathbf{w}) \cdot (a\mathbf{w}) = 0$ for every scalar a. The dot product has the property that $(\mathbf{x} - c\mathbf{w}) \cdot (a\mathbf{w}) = a((\mathbf{x} - c\mathbf{w}) \cdot \mathbf{w})$, so actually it suffices to make sure that $(\mathbf{x} - c\mathbf{w}) \cdot \mathbf{w} = 0$. We can use the further properties of the dot product to rewrite this as

$$0 = (\mathbf{x} - c\mathbf{w}) \cdot \mathbf{w} = \mathbf{x} \cdot \mathbf{w} - (c\mathbf{w}) \cdot \mathbf{w}.$$

We can rearrange this expression to write it as $c(\mathbf{w} \cdot \mathbf{w}) = \mathbf{x} \cdot \mathbf{w}$. But $\mathbf{w} \cdot \mathbf{w} = \|\mathbf{w}\|^2 > 0$ (since $\mathbf{w} \neq \mathbf{0}$), so it makes sense to divide both sides by $\mathbf{w} \cdot \mathbf{w}$ to obtain that $c = \dfrac{\mathbf{x} \cdot \mathbf{w}}{\mathbf{w} \cdot \mathbf{w}}$. This is the coefficient we had previously regarded as unknown! To summarize, we have shown through algebra and the properties of dot products that there is exactly one point in the line $L = \operatorname{span}(\mathbf{w})$ through $\mathbf{0}$ in $\mathbf{R}^n$ whose difference from $\mathbf{x}$ is orthogonal to everything in L: it is $\left(\dfrac{\mathbf{x} \cdot \mathbf{w}}{\mathbf{w} \cdot \mathbf{w}}\right) \mathbf{w}$.

We have not yet actually shown that this scalar multiple of $\mathbf{w}$ on L is closer to $\mathbf{x}$ than every other vector in L, but if we believe the orthogonality insight inspired by the 2-dimensional picture in Figure 6.1.1 then this must be that closest point. As a bonus, we have obtained an explicit formula for it!

Method II (geometric). We next use some plane geometry via Figure 6.1.1 to obtain the same formula for the closest point. Strictly speaking, this argument only applies when $n = 2$, but you might find that it gives the formula some visual meaning that is somehow lacking in the purely algebraic work in Method I. There is nothing to be done if $\mathbf{x} \cdot \mathbf{w} = \mathbf{0}$ (in that case $\mathbf{x}$ is perpendicular to L and we are thereby convinced that $\mathbf{0}$ is the closest point, as is also given by the desired formula). Hence, we can suppose $\mathbf{x} \neq \mathbf{0}$ and the angle θ between $\mathbf{x}$ and $\mathbf{w}$ satisfies either $0° < \theta < 90°$ or $90° < \theta < 180°$.

The case of acute θ is shown in Figure 6.1.1, whereas if θ is obtuse then the point we seek would be in the direction of $-\mathbf{w}$ (rather than in the direction of $\mathbf{w}$). If θ is acute, as in Figure 6.1.1, then by basic trigonometry, the leg along L for the right triangle as shown has length $\|\mathbf{x}\| \cos(\theta)$ and it points in the direction of the unit vector $\mathbf{w}/\|\mathbf{w}\|$. This says that the endpoint on L of the dotted segment is the vector

$$(\|\mathbf{x}\| \cos(\theta)) \frac{\mathbf{w}}{\|\mathbf{w}\|}. \tag{6.1.1}$$

But $\cos(\theta) = (\mathbf{x} \cdot \mathbf{w})/(\|\mathbf{x}\| \|\mathbf{w}\|)$, so plugging this into (6.1.1) yields the desired formula since $\|\mathbf{w}\|^2 = \mathbf{w} \cdot \mathbf{w}$. The case when $90^\circ < \theta < 180^\circ$ goes very similarly, except now $\cos(\theta) < 0$ (so the endpoint on L of the dotted segment is in the direction of the opposite unit vector $-\mathbf{w}/\|\mathbf{w}\|$) and we have to work with the length $\|\mathbf{x}\| \, |\cos(\theta)| = -\|\mathbf{x}\| \cos(\theta)$. Putting these together, the two signs cancel and we get the desired formula again.

The following result records the conclusion of our preceding algebraic and geometric considerations, inspired by Figure 6.1.1. For those who wish to think about this more carefully, we provide at the end of this section a logically complete proof that works directly in every $\mathbf{R}^n$ (and shows in particular that the insight about orthogonality we have been using is always correct).

Proposition 6.1.1. Let $L = \operatorname{span}(\mathbf{w}) = \{c\mathbf{w} : c \in \mathbf{R}\}$ be a 1-dimensional linear subspace of $\mathbf{R}^n$ (so $\mathbf{w} \neq \mathbf{0}$), a "line". Choose any point $\mathbf{x} \in \mathbf{R}^n$. There is exactly one point in L closest to $\mathbf{x}$, and it is given by the scalar multiple

$$\left(\frac{\mathbf{x} \cdot \mathbf{w}}{\mathbf{w} \cdot \mathbf{w}}\right) \mathbf{w} \tag{6.1.2}$$

of $\mathbf{w}$. This is called "the projection of $\mathbf{x}$ into $\operatorname{span}(\mathbf{w})$"; we denote it by the symbol $\mathbf{Proj}_{\mathbf{w}}\, \mathbf{x}$.

WARNING: Make sure you understand the notation in the formula for this closest point, since it is mixing scalars – the ratio of dot products – and vectors.

Example 6.1.2. For the line L in $\mathbf{R}^2$ through the origin given by $y = 2x$, let us compute the projection of $\mathbf{x} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}$ onto L. The line L is spanned by any nonzero vector in L, such as $\mathbf{w} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ (by setting $x = 1$ in the equation for L), so (6.1.2) gives that the projection of $\mathbf{x}$ onto L is

$$\mathbf{Proj}_{\mathbf{w}}\,\mathbf{x} = \left(\frac{\mathbf{x}\cdot\mathbf{w}}{\mathbf{w}\cdot\mathbf{w}}\right)\mathbf{w} = \frac{7}{5}\mathbf{w} = \begin{bmatrix} 7/5 \\ 14/5 \end{bmatrix} = \begin{bmatrix} 1.4 \\ 2.8 \end{bmatrix}$$

as illustrated in Figure 6.1.2.

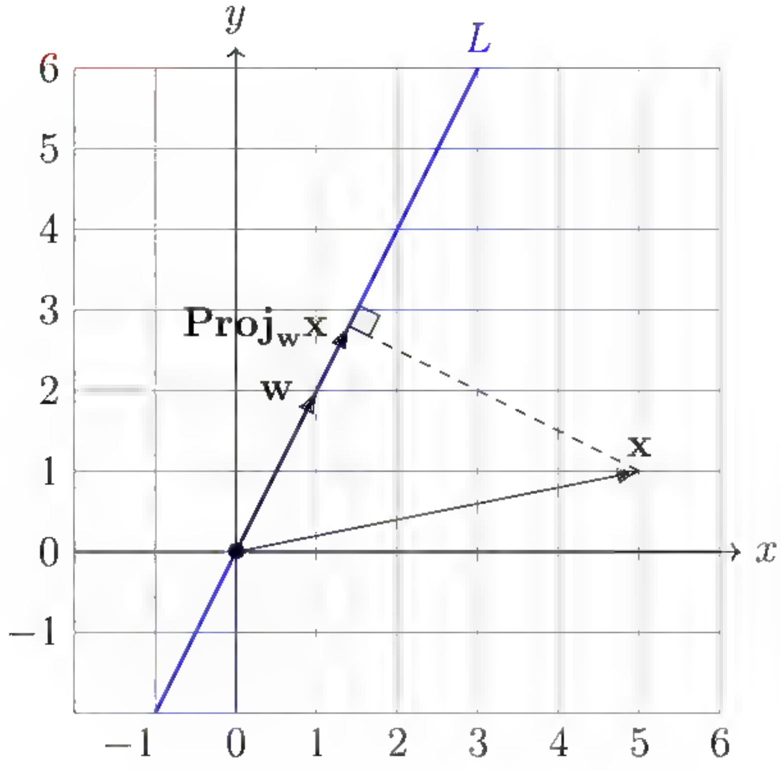


FIGURE 6.1.2. Projection of the vector $(5, 1)$ onto the line $y = 2x$

It is a good idea to check that, as in the general picture in Figure 6.1.1, the difference vector

$$\mathbf{x} - \mathbf{Proj}_{\mathbf{w}}\,\mathbf{x} = \begin{bmatrix} 5 \\ 1 \end{bmatrix} - \begin{bmatrix} 7/5 \\ 14/5 \end{bmatrix} = \begin{bmatrix} 18/5 \\ -9/5 \end{bmatrix}$$

(this is the dotted segment in Figure 6.1.2 up to a choice of direction) is indeed perpendicular to the line L, or equivalently is perpendicular to the nonzero $\mathbf{w}$ that spans L. The dot product of this difference vector with $\mathbf{w}$ is $1 \cdot (18/5) + 2(-9/5) = 0$. ■

Example 6.1.3. In many engineering problems involving a balance of forces (such as to determine the tension in a system of cables used to lift a heavy beam into the air, or a myriad of systems involving the support of objects by a linked system of springs, chains, cords, etc.), a key part of the analysis is to determine the projection of a force vector along various directions relevant to the geometry of the situation.

For example, on a hill that makes an angle of 25° relative to the ground, suppose a heavy sled is being pulled by a rope with a force of 200 Newtons. Depending on the height of the person, the taut rope will make some further angle relative to the slope of the hill; say it is 50°. Focusing only on the force applied by the person dragging the sled up the hill, the force vector $\mathbf{F}$ thereby points in the direction of that 50° incline relative to the hill, as shown in Figure 6.1.3.

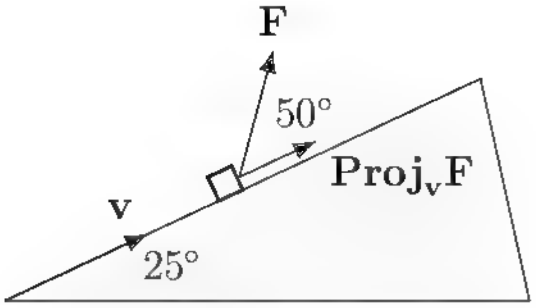


FIGURE 6.1.3. Pulling a heavy object up a hill

The projection of $\mathbf{F}$ along the direction of the hill is the part of the force that contributes to actual motion (sometimes called the "effective force") and the projection in the direction perpendicular to the hill is "wasted effort" (say ignoring friction).

If $\mathbf{v}$ is a vector pointing up the direction of the hill then to find the "effective force" we want to compute the magnitude of $\mathbf{Proj_v}(\mathbf{F})$. This projection is

$$\left(\frac{\mathbf{v}\cdot\mathbf{F}}{\mathbf{v}\cdot\mathbf{v}}\right)\mathbf{v}=\left(\frac{\|\mathbf{v}\|\|\mathbf{F}\|\cos\theta}{\|\mathbf{v}\|^2}\right)\mathbf{v}=\left(\frac{\|\mathbf{F}\|\cos\theta}{\|\mathbf{v}\|}\right)\mathbf{v}$$

where $\|\mathbf{F}\|=200$ is the magnitude (in Newtons) of the applied force and the angle θ between $\mathbf{F}$ and $\mathbf{v}$ is $50°$. Hence, the projection $\mathbf{Proj_v}(\mathbf{F})$ is $200\cos(50°)(\mathbf{v}/\|\mathbf{v}\|)$. The vector $\mathbf{v}/\|\mathbf{v}\|$ has length 1 since it is a vector divided by its length. Hence, when we compute the magnitude of this projection what remains is the multiplier $200\cos(50°)\approx 128.56$ Newtons. Note that the angle of the hill has *nothing* to do with this calculation (though it impacts the roles of friction and gravity in more thorough calculations).

Similarly, for any magnitude F of applied force to pull the sled up the hill with the taut rope making an angle of θ relative to the hill, the "effective force" applied is $F\cos\theta$ (if we ignore friction). ■

Example 6.1.4. Let us now find the closest point to $\mathbf{x}=\begin{bmatrix}1\\2\\4\end{bmatrix}$ on the line $\mathrm{span}(\mathbf{w})$ for $\mathbf{w}=\begin{bmatrix}3\\-1\\1\end{bmatrix}$, illustrated in Figure 6.1.4.

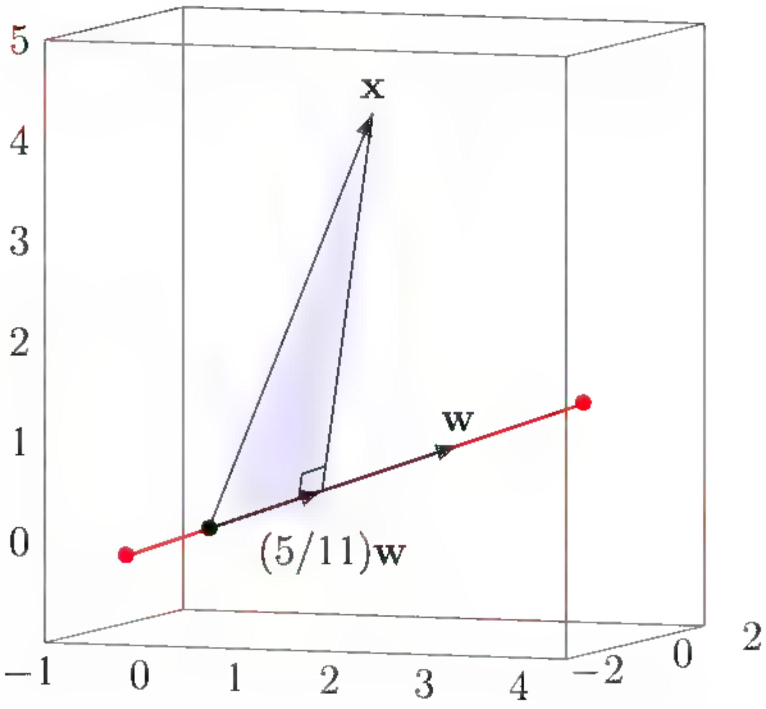


FIGURE 6.1.4. The orthogonal projection of $\mathbf{x}$ onto the span of $\mathbf{w}$ is $(5/11)\mathbf{w}$

Since $\mathbf{w}\cdot\mathbf{w} = 9+1+1 = 11$ and $\mathbf{x}\cdot\mathbf{w} = 3-2+4 = 5$ we compute $\mathbf{Proj}_{\mathbf{w}}(\mathbf{x}) = ((\mathbf{x}\cdot\mathbf{w})/(\mathbf{w}\cdot\mathbf{w}))\mathbf{w} = (5/11)\mathbf{w} = \begin{bmatrix} 15/11 \\ -5/11 \\ 5/11 \end{bmatrix}$ as in Figure 6.1.4. Thus, the nearest point on the line to $\begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$ is $\begin{bmatrix} 15/11 \\ -5/11 \\ 5/11 \end{bmatrix}$. ■

There is a very useful algebraic property of projections onto lines L through $\mathbf{0}$ that is difficult to "see" directly in purely geometric terms but is quite convenient when one needs to compute $\mathbf{Proj}_L(\mathbf{v})$ for many different points $\mathbf{v}$. This is a nice synthesis of algebraic and geometric aspects of linear algebra. The upshot is going to be that if we do the work of computing the vectors $\mathbf{Proj}_L(\mathbf{e}_i)$ for all $i = 1, \dots, n$ and store those in a computer, then that data can be used to *very rapidly* compute $\mathbf{Proj}_L(\mathbf{v})$ for any vector $\mathbf{v} \in \mathbf{R}^n$ whatsoever.

We illustrate this with a specific example. Consider $\mathbf{v} = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}$. The numbers $1, 3, 4$ represent *the amount of* $\mathbf{v}$ *that points along the* x, y, z*-axes respectively.* More precisely, $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{e}_1$ is the component of $\mathbf{v}$ along the x-axis line, $\begin{bmatrix} 0 \\ 3 \\ 0 \end{bmatrix} = 3\mathbf{e}_2$ is the component of $\mathbf{v}$ along the y-axis line, and $\begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix} = 4\mathbf{e}_3$ is the component of $\mathbf{v}$ along the z-axis line. (These are the closest points to $\mathbf{v}$ on the x-, y- and z-axes, respectively.)

In terms of this data, we want to compute the projection of $\mathbf{v}$ on some line pointing with *some other direction*: if $\mathbf{w}$ is a (nonzero) vector along this new direction, we want to compute $\mathbf{Proj}_{\mathbf{w}}(\mathbf{v})$. The key point is that the formula (6.1.2) for $\mathbf{Proj}_{\mathbf{w}}(\mathbf{x})$ behaves well for any linear combination of any n-vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$: the projection of a linear combination of the $\mathbf{x}_i$'s is equal to the corresponding linear combination of the projections.

For example, with $k = 2$ it says $\mathbf{Proj}_{\mathbf{w}}(5\mathbf{x}_1 - 7\mathbf{x}_2) = 5\,\mathbf{Proj}_{\mathbf{w}}(\mathbf{x}_1) - 7\,\mathbf{Proj}_{\mathbf{w}}(\mathbf{x}_2)$ and likewise with 5 and -7 replaced by any two scalars. The reason this works is an algebraic calculation:

$$\begin{aligned}
\mathbf{Proj}_{\mathbf{w}}(c_1\mathbf{x}_1 + \cdots + c_k\mathbf{x}_k) &= \left(\frac{(c_1\mathbf{x}_1 + \cdots + c_k\mathbf{x}_k)\cdot\mathbf{w}}{\mathbf{w}\cdot\mathbf{w}}\right)\mathbf{w} \\
&= \left(\frac{c_1(\mathbf{x}_1\cdot\mathbf{w}) + \cdots + c_k(\mathbf{x}_k\cdot\mathbf{w})}{\mathbf{w}\cdot\mathbf{w}}\right)\mathbf{w} \\
&= c_1\left(\frac{\mathbf{x}_1\cdot\mathbf{w}}{\mathbf{w}\cdot\mathbf{w}}\right)\mathbf{w} + \cdots + c_k\left(\frac{\mathbf{x}_k\cdot\mathbf{w}}{\mathbf{w}\cdot\mathbf{w}}\right)\mathbf{w} \\
&= c_1\,\mathbf{Proj}_{\mathbf{w}}(\mathbf{x}_1) + \cdots + c_k\,\mathbf{Proj}_{\mathbf{w}}(\mathbf{x}_k). \qquad (6.1.3)
\end{aligned}$$

The fact that $\mathbf{Proj}_{\mathbf{w}}(\cdot)$ behaves so nicely with respect to linear combinations will be *very useful* but is not at all apparent in terms of the "nearest point" definition of $\mathbf{Proj}_{\mathbf{w}}(\cdot)$. It dropped out from the formula for $\mathbf{Proj}_{\mathbf{w}}(\cdot)$ in terms of dot products in (6.1.2), demonstrating the power of combining algebraic and geometric ideas in linear algebra. Applying (6.1.3) to the expression $\mathbf{v} = \mathbf{e}_1 + 3\mathbf{e}_2 + 4\mathbf{e}_3$ yields

$$\mathbf{Proj}_{\mathbf{w}}(\mathbf{v}) = \mathbf{Proj}_{\mathbf{w}}(\mathbf{e}_1) + 3\,\mathbf{Proj}_{\mathbf{w}}(\mathbf{e}_2) + 4\,\mathbf{Proj}_{\mathbf{w}}(\mathbf{e}_3). \qquad (6.1.4)$$

A visualization of (6.1.4) is given in Figure 6.1.5: the projections $\mathbf{e}_1, 3\mathbf{e}_2, 4\mathbf{e}_3$ of $\mathbf{v}$ into the coordinate axes are indicated with green, the projections of those three into the blue line $L = \operatorname{span}(\mathbf{w})$ are indicated with purple, and the sum of those latter three projections in L is equal to $\mathbf{Proj}_{\mathbf{w}}(\mathbf{v})$. The picture looks complicated, so the equality (6.1.4) may be surprising (or confusing) when thinking about it in purely geometric terms via distance minimization, but via vector algebra as above it was a direct calculation.

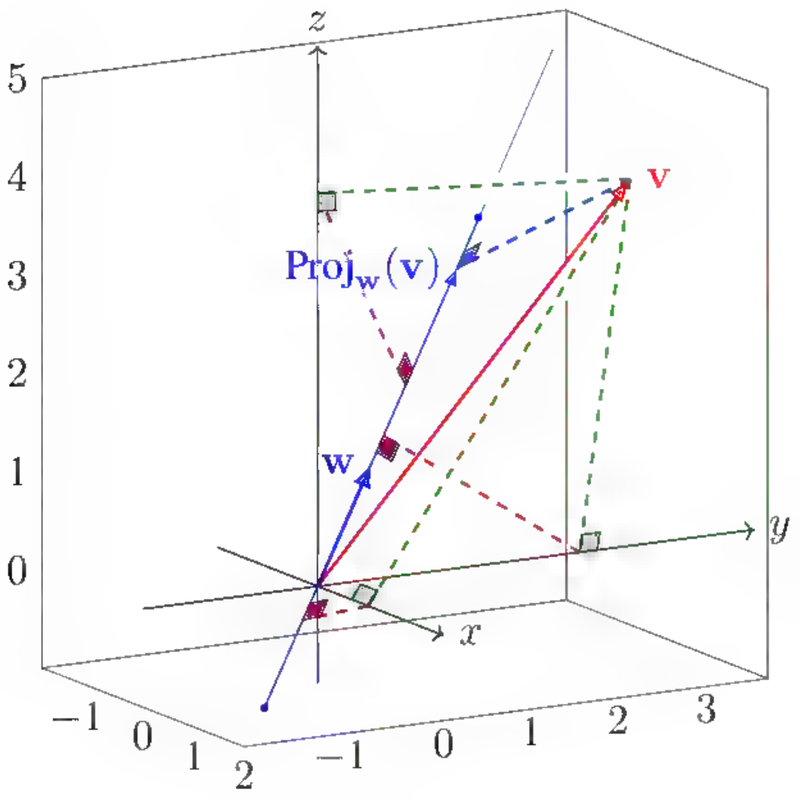


FIGURE 6.1.5. The projection $\mathbf{Proj}_{\mathbf{w}}(\mathbf{v})$ in the blue line of the red vector $\mathbf{v}$ is the *sum* of the 3 purple projections in the blue line of the 3 green projections of $\mathbf{v}$ into the coordinate axes. This illustrates how surprising (6.1.4) is as a geometric fact (without algebra).

The lesson is that we can compute the projection of $\mathbf{v}$ into an *arbitrary* line $L = \mathrm{span}(\mathbf{w})$ through $\mathbf{0}$ in terms of the components 1, 3, 4 of $\mathbf{v}$ along the coordinate axes, namely as the "same" linear combination of the projections of $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ into the same line L. Let's see this in action, to appreciate its usefulness.

Example 6.1.5. Let $\mathbf{v} = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}$ be as above. Consider the following line in $\mathbf{R}^3$ through the origin:

$$L = \left\{ \begin{bmatrix} t \\ 0 \\ t \end{bmatrix} : t \in \mathbf{R} \right\}.$$

We shall compute in *two ways* the point on L that is closest to $\mathbf{v}$. Begin by picking a nonzero vector $\mathbf{w}$ on L, say $\mathbf{w} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$. Whatever the choice of $\mathbf{w}$ we make, since L is a line we have $L = \mathrm{span}(\mathbf{w})$ and $\mathbf{Proj}_L(\mathbf{v}) = \mathbf{Proj}_{\mathbf{w}}(\mathbf{v})$. Our task is to calculate $\mathbf{Proj}_{\mathbf{w}}(\mathbf{v})$.

Method I. By Proposition 6.1.1 we have

$$\mathbf{Proj}_L(\mathbf{v}) = \mathbf{Proj}_{\mathbf{w}}(\mathbf{v}) = \left(\frac{\mathbf{v} \cdot \mathbf{w}}{\mathbf{w} \cdot \mathbf{w}} \right) \mathbf{w}. \tag{6.1.5}$$

To compute this, we first work out some dot products: $\mathbf{v} \cdot \mathbf{w} = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = 5$ and $\mathbf{w} \cdot \mathbf{w} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = 2$.

Hence, by (6.1.5), $\mathbf{Proj}_L(\mathbf{v}) = \dfrac{5}{2} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 5/2 \\ 0 \\ 5/2 \end{bmatrix}$.

Method II. Based on (6.1.4), we first compute the projections $\mathbf{Proj}_{\mathbf{w}}(\mathbf{e}_i)$. Using the formula from Proposition 6.1.1, we have

$$\mathbf{Proj}_{\mathbf{w}}(\mathbf{e}_i) = \left(\frac{\mathbf{e}_i \cdot \mathbf{w}}{\mathbf{w} \cdot \mathbf{w}}\right) \mathbf{w} = \left(\frac{\mathbf{e}_i \cdot \mathbf{w}}{2}\right) \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

with $\mathbf{e}_i \cdot \mathbf{w}$ equal to the ith entry in $\mathbf{w}$. In other words:

$$\mathbf{Proj}_{\mathbf{w}}(\mathbf{e}_1) = \frac{1}{2}\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 0 \\ 1/2 \end{bmatrix}, \quad \mathbf{Proj}_{\mathbf{w}}(\mathbf{e}_2) = \frac{0}{2}\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{Proj}_{\mathbf{w}}(\mathbf{e}_3) = \frac{1}{2}\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 0 \\ 1/2 \end{bmatrix},$$

so (6.1.4) yields

$$\mathbf{Proj}_{\mathbf{w}}(\mathbf{v}) = \begin{bmatrix} 1/2 \\ 0 \\ 1/2 \end{bmatrix} + 3\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + 4\begin{bmatrix} 1/2 \\ 0 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 5/2 \\ 0 \\ 5/2 \end{bmatrix}$$

once again.

You might think that Method II is silly, since it involves applying Proposition 6.1.1 three times (once for each $\mathbf{Proj}_{\mathbf{w}}(\mathbf{e}_i)$) rather than once as in Method I. The merit of Method II is seen if we have to compute $\mathbf{Proj}_{\mathbf{w}}(\mathbf{x})$ for *lots* of different vectors $\mathbf{x}$. Namely, the procedure yielding (6.1.4) carries over to give a completely general formula for $\mathbf{Proj}_{\mathbf{w}}(\mathbf{x})$ in terms of the entries of $\mathbf{x}$: if $\mathbf{x} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ then

$$\mathbf{Proj}_{\mathbf{w}}(\mathbf{x}) = a\,\mathbf{Proj}_{\mathbf{w}}(\mathbf{e}_1) + b\,\mathbf{Proj}_{\mathbf{w}}(\mathbf{e}_2) + c\,\mathbf{Proj}_{\mathbf{w}}(\mathbf{e}_3),$$

so once we have done the work to compute the three vectors $\mathbf{Proj}_{\mathbf{w}}(\mathbf{e}_i)$ (as we did above) then this yields a general formula for *all* $\mathbf{x}$:

$$\mathbf{Proj}_{\mathbf{w}}(\mathbf{x}) = a\begin{bmatrix} 1/2 \\ 0 \\ 1/2 \end{bmatrix} + b\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + c\begin{bmatrix} 1/2 \\ 0 \\ 1/2 \end{bmatrix} = \begin{bmatrix} (a+c)/2 \\ 0 \\ (a+c)/2 \end{bmatrix}.$$

For any nonzero $\mathbf{w} \in \mathbf{R}^n$, Method II adapts to any $\mathbf{R}^n$ to give a general formula for $\mathbf{Proj}_{\mathbf{w}}(\mathbf{x})$ in terms of the entries of a general $\mathbf{x} \in \mathbf{R}^n$ once we have computed the $\mathbf{Proj}_{\mathbf{w}}(\mathbf{e}_i)$'s (say by applying Proposition 6.1.1 for each $i = 1, \dots, n$). ■

Example 6.1.6. In Example 6.1.3, we saw that projection to a line arises in the analysis of forces. Projection to lines also arises, for entirely different reasons, at the heart of quantum mechanics. In that physical theory, which underlies all modern electronics, the possible states of a (microscopic) system correspond to unit vectors $\mathbf{x}$ in a typically large-dimensional "state space".

For each "observable" quantity of interest, there is an orthonormal basis $\mathbf{v}_1, \mathbf{v}_2, \dots$ of the state space so that the $\mathbf{v}_j$'s correspond to the possible outcomes of a measurement on $\mathbf{x}$ of the observable quantity. An important discovery of Max Born is that $\|\mathbf{Proj}_{\mathbf{v}_j}(\mathbf{x})\|^2$ is the probability that a measurement of the observable for a system in state $\mathbf{x}$ has outcome corresponding to $\mathbf{v}_j$. The Fourier formula (5.3.5) applied to $\mathbf{x}$ says $\mathbf{x} = \sum_j \mathbf{Proj}_{\mathbf{v}_j}(\mathbf{x})$; this is a sum of *pairwise orthogonal* vectors, and a generalization of the Pythagorean Theorem to sums of pairwise orthogonal vectors (the case of a sum of two such vectors is Theorem 2.3.1) then gives $\|\mathbf{x}\|^2 = \sum_j \|\mathbf{Proj}_{\mathbf{v}_j}(\mathbf{x})\|^2$. But $\|\mathbf{x}\| = 1$ since $\mathbf{x}$ is a unit vector, so this says the sum of the probabilities of all possible outcomes is 1, as it ought to be. ■

PROOF OF PROPOSITION 6.1.1.

We are seeking a point of the form $t\mathbf{w}$ with minimal distance to $\mathbf{x}$: the difference vector $\mathbf{x} - t\mathbf{w}$ has minimal length. Our aim is to show that $\mathbf{x} - t\mathbf{w}$ has minimal length precisely when $\mathbf{x} - t\mathbf{w}$ is perpendicular to $\mathbf{w}$, and that in this case $t = (\mathbf{x} \cdot \mathbf{w})/(\mathbf{w} \cdot \mathbf{w})$ as in (6.1.2). Finding the "closest point" to the line might seem like a calculus problem: set up a distance function in terms of t and minimize it. This can be done via calculus, but it is *much easier* (and more illuminating) to solve this with a bit of linear algebra, as we shall now see. First, we check that the *only* value of t for which $\mathbf{x} - t\mathbf{w}$ is perpendicular to $\mathbf{w}$ is the coefficient in (6.1.2). We compute $(\mathbf{x} - t\mathbf{w}) \cdot \mathbf{w} = (\mathbf{x} \cdot \mathbf{w}) - (t\mathbf{w}) \cdot \mathbf{w} = (\mathbf{x} \cdot \mathbf{w}) - t(\mathbf{w} \cdot \mathbf{w})$, so this vanishes precisely when $t(\mathbf{w} \cdot \mathbf{w}) = \mathbf{x} \cdot \mathbf{w}$. Dividing by $\mathbf{w} \cdot \mathbf{w} = \|\mathbf{w}\|^2 > 0$, this is the same as saying $t = (\mathbf{x} \cdot \mathbf{w})/(\mathbf{w} \cdot \mathbf{w})$, as desired.

It remains to show that the distance $\|\mathbf{x} - t\mathbf{w}\| \geq 0$ is minimized exactly when $t = (\mathbf{x} \cdot \mathbf{w})/(\mathbf{w} \cdot \mathbf{w})$. It is the same to show $\|\mathbf{x} - t\mathbf{w}\|^2$ is minimized exactly when t has that value. Similarly to what we saw in Section 2.3, $\|\mathbf{x} - t\mathbf{w}\|^2 = (\mathbf{x} - t\mathbf{w}) \cdot (\mathbf{x} - t\mathbf{w}) = \mathbf{x} \cdot \mathbf{x} - 2t(\mathbf{x} \cdot \mathbf{w}) + t^2(\mathbf{w} \cdot \mathbf{w}) = (\mathbf{w} \cdot \mathbf{w})t^2 - 2(\mathbf{x} \cdot \mathbf{w})t + \mathbf{x} \cdot \mathbf{x}$. This is a quadratic polynomial in t with leading coefficient $\mathbf{w} \cdot \mathbf{w} = \|\mathbf{w}\|^2$ that is *positive*. But any quadratic polynomial $at^2 + bt + c$ with $a > 0$ has parabolic graph with minimum exactly at the vertex $t = -b/(2a)$. In our case $a = \mathbf{w} \cdot \mathbf{w}$ and $b = -2(\mathbf{x} \cdot \mathbf{w})$, so $-b/(2a) = (\mathbf{x} \cdot \mathbf{w})/(\mathbf{w} \cdot \mathbf{w})$.

6.2. Projection onto a general subspace. We have now acquired a lot of experience with projection to lines through $\mathbf{0}$ in $\mathbf{R}^n$. Let us harness that skill to compute the point in a linear subspace V of $\mathbf{R}^n$ nearest to a chosen $\mathbf{x} \in \mathbf{R}^n$, which we denote as $\mathbf{Proj}_V(\mathbf{x}) \in V$. This nearest point to $\mathbf{x}$ will be computed as a sum of projections of $\mathbf{x}$ into lines through $\mathbf{0}$ in V arising from an orthogonal basis of V (when V is nonzero).

The first issue is to adapt to general V the orthogonality insight we obtained in Figure 6.1.1 with lines; we used that to obtain a formula for the nearest point to $\mathbf{x}$ on a line through the origin in $\mathbf{R}^n$. As motivation in general, let's look at the case of a plane V through the origin in $\mathbf{R}^3$ equipped with a choice of orthogonal basis $\{\mathbf{v}_1, \mathbf{v}_2\}$ of this plane. In Figure 6.2.1 we draw the typical situation, indicating with the notation $\mathbf{Proj}_V(\mathbf{x})$ the point in V closest to $\mathbf{x}$. The first geometric insight, similar to our experience with lines, is that since this nearest point should have displacement vector to $\mathbf{x}$ that is the "most direct" route to V from $\mathbf{x}$, the displacement should involve "no tilting" relative to any direction *within* V.

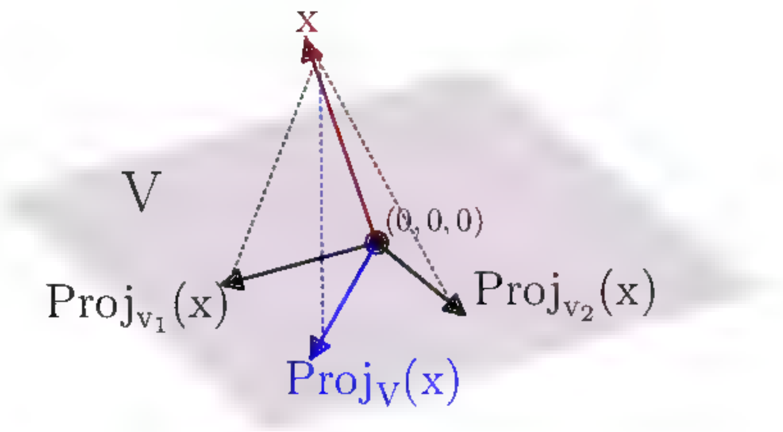


FIGURE 6.2.1. The case $n = 3$ with V a plane: $\mathbf{Proj}_V(\mathbf{x}) = \mathbf{Proj}_{\mathbf{v}_1}(\mathbf{x}) + \mathbf{Proj}_{\mathbf{v}_2}(\mathbf{x})$.

In Figure 6.2.1, although the displacement $\mathbf{x} - \mathbf{Proj}_{\mathbf{v}_i}(\mathbf{x})$ to the line $\mathrm{span}(\mathbf{v}_i)$ is perpendicular to that line, it is generally *not* perpendicular to much else in V. This is indicated by the tilt relative to V of the triangle with vertices $(0, 0, 0)$, $\mathbf{x}$, and $\mathbf{Proj}_{\mathbf{v}_2}(\mathbf{x})$ in Figure 6.2.1. The triangle with vertices $(0, 0, 0)$, $\mathbf{x}$, and

$\mathbf{Proj}_{\mathbf{v}_1}(\mathbf{x})$ is also tilted (perhaps harder to notice, due to perspective), whereas the triangle with vertices $(0, 0, 0)$, $\mathbf{x}$, and $\mathbf{Proj}_V(\mathbf{x})$ has *no tilt* relative to V. This contrast of tilting and non-tilting triangles relative to V expresses that the displacement $\mathbf{x} - \mathbf{Proj}_V(\mathbf{x})$ is perpendicular to *everything* in V.

If you think about it, hopefully it seems plausible that if $\mathbf{v} \in V$ makes the displacement $\mathbf{x} - \mathbf{v}$ perpendicular to everything in V then $\mathbf{v}$ should be the point in V for which the direction of the displacement $\mathbf{x} - \mathbf{v}$ is the "most direct" route from $\mathbf{x}$ to V, making $\mathbf{v}$ the point in V nearest to $\mathbf{x}$. The picture in Figure 6.2.1 suggests that the rectangle having as two of its edges the segments joining $(0, 0, 0)$ to $\mathbf{Proj}_{\mathbf{v}_1}(\mathbf{x})$ and $\mathbf{Proj}_{\mathbf{v}_2}(\mathbf{x})$ has as its corner opposite $(0, 0, 0)$ the point $\mathbf{Proj}_V(\mathbf{x})$. By the parallelogram law for vector addition, this would say $\mathbf{Proj}_V(\mathbf{x}) = \mathbf{Proj}_{\mathbf{v}_1}(\mathbf{x}) + \mathbf{Proj}_{\mathbf{v}_2}(\mathbf{x})$.

Here is the general result encompassing our observations in the special case $n = 3$ with V a plane:

Theorem 6.2.1 (Orthogonal Projection Theorem, version I). For any $\mathbf{x} \in \mathbf{R}^n$ and linear subspace V of $\mathbf{R}^n$, there is a unique $\mathbf{v}$ in V closest to $\mathbf{x}$. (In symbols, $\|\mathbf{x} - \mathbf{v}\| < \|\mathbf{x} - \mathbf{v}'\|$ for all $\mathbf{v}'$ in V with $\mathbf{v}' \neq \mathbf{v}$.) This $\mathbf{v}$ is called the *projection of* $\mathbf{x}$ *onto* V, and is denoted $\mathbf{Proj}_V(\mathbf{x})$; see Figure 6.2.1. The projection $\mathbf{Proj}_V(\mathbf{x})$ is also the *only* vector $\mathbf{v} \in V$ with the property that the displacement $\mathbf{x} - \mathbf{v}$ is perpendicular to V (i.e., $\mathbf{x} - \mathbf{v}$ is perpendicular to *every* vector in V).

If V is nonzero then for any orthogonal basis $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ of V we have

$$\mathbf{Proj}_V(\mathbf{x}) = \mathbf{Proj}_{\mathbf{v}_1}(\mathbf{x}) + \mathbf{Proj}_{\mathbf{v}_2}(\mathbf{x}) + \cdots + \mathbf{Proj}_{\mathbf{v}_k}(\mathbf{x}), \tag{6.2.1}$$

where $\mathbf{Proj}_{\mathbf{v}_i}(\mathbf{x}) = ((\mathbf{x} \cdot \mathbf{v}_i)/(\mathbf{v}_i \cdot \mathbf{v}_i))\mathbf{v}_i$ as in Proposition 6.1.1. (For $\mathbf{x} \in V$ we have $\mathbf{Proj}_V(\mathbf{x}) = \mathbf{x}$ – the point in V closest to $\mathbf{x}$ is itself! – so (6.2.1) for $\mathbf{x} \in V$ recovers (5.3.5)!)

When $\dim V = 1$, the link between perpendicularity and distance minimization was discussed in Section 6.1. In Section 6.1 we saw that the orthogonality characterization of the nearest point to $\mathbf{x}$ on a line through $\mathbf{0}$ was the key to actually computing that point, and the same goes for the general case in Theorem 6.2.1. A proof of Theorem 6.2.1 for general V is given in Section 6.3.

Although (6.2.1) in the special case $\mathbf{x} \in V$ is exactly (5.3.5), it may still seem a bit unintuitive why we should expect (6.2.1) to be true in general. We give some motivation for this in Remark 6.2.5. That motivation gives some insight into why it is necessary to assume that the basis $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ of V is *orthogonal* in order for the formula (6.2.1) to hold. A way to see that one cannot expect (6.2.1) to hold for a general basis of V is to revisit the picture in Figure 6.2.1 but with $\mathbf{v}_2$ taken to be extremely close to $\mathbf{v}_1$: in that case $\mathbf{Proj}_{\mathbf{v}_2}(\mathbf{x})$ would be extremely close to $\mathbf{Proj}_{\mathbf{v}_1}(\mathbf{x})$ and so their sum would be extremely close to $2\,\mathbf{Proj}_{\mathbf{v}_1}(\mathbf{x})$, which has nothing at all to do with the point $\mathbf{Proj}_V(\mathbf{x})$ in V closest to $\mathbf{x}$!

Example 6.2.2. According to [CT] and the related press release [Daj], also described for the general public in [T, pp. 27-28] (see the sections "Cracking the Code" and "A Win-Win Bet"), facial recognition in the brain amounts to computing a "face vector" $\mathbf{x} \in \mathbf{R}^{50}$ by using (6.2.1) for $V = \mathbf{R}^n = \mathbf{R}^{50}$!

Techniques from Principal Component Analysis (see Section 27.3) provide a special orthonormal basis $\mathbf{v}_1, \ldots, \mathbf{v}_{50}$ of $\mathbf{R}^{50}$ (**not** the standard basis) so that $\mathbf{Proj}_{\mathbf{v}_j}(\mathbf{x})$ is computed by specific neurons depending on j; the brain sums these to compute $\mathbf{Proj}_{\mathbf{R}^{50}}(\mathbf{x}) = \mathbf{x}$. In [T], the role of projections in this breakthrough is described by the lead scientist, Doris Y. Tsao, as follows:

> "Remembering my ... linear algebra, I realized ... that we should be able to construct a large "null space" of faces for each face cell ... It would demolish the vague intuition that everyone shared about face cells – that they should be tuned to specific faces ... face cells are not encoding the identities of specific individuals in the IT cortex. Instead, they are performing an axis projection, a much more abstract computation." ■

Example 6.2.3. Let U be the subspace of $\mathbf{R}^4$ spanned by $\mathbf{u}_1 = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}$ and $\mathbf{u}_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \end{bmatrix}$. Let $\mathbf{v} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$. We seek the point $\mathbf{u}$ in U that is closest to $\mathbf{v}$.

To solve this problem, we want to use Theorem 6.2.1 to find the projection $\mathbf{Proj}_U(\mathbf{v})$. Since

$$\mathbf{u}_1 \cdot \mathbf{u}_2 = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \end{bmatrix} = 0,$$

the nonzero $\mathbf{u}_1$ and $\mathbf{u}_2$ form an orthogonal basis for U (so U is a "plane"; i.e., $\dim(U) = 2$). Thus, Theorem 6.2.1 tells us that $\mathbf{u} = \mathbf{Proj}_U(\mathbf{v}) = \mathbf{Proj}_{\mathbf{u}_1}(\mathbf{v}) + \mathbf{Proj}_{\mathbf{u}_2}(\mathbf{v})$.

These projections are given by $\mathbf{Proj}_{\mathbf{u}_1}(\mathbf{v}) = c_1\mathbf{u}_1$ and $\mathbf{Proj}_{\mathbf{u}_2}(\mathbf{v}) = c_2\mathbf{u}_2$ for the coefficients

$$c_1 = \frac{\mathbf{v} \cdot \mathbf{u}_1}{\mathbf{u}_1 \cdot \mathbf{u}_1}, \quad c_2 = \frac{\mathbf{v} \cdot \mathbf{u}_2}{\mathbf{u}_2 \cdot \mathbf{u}_2}.$$

To evaluate these, we need to compute some dot products. By inspection $\mathbf{v} \cdot \mathbf{u}_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} = -1$ and $\mathbf{u}_1 \cdot \mathbf{u}_1 = 1^2 + (-1)^2 = 2$, so $c_1 = -1/2$. Likewise, $\mathbf{v} \cdot \mathbf{u}_2 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \end{bmatrix} = 14$ and $\mathbf{u}_2 \cdot \mathbf{u}_2 = 1^2 + 1^2 + 1^2 + 2^2 = 7$, so $c_2 = 14/7 = 2$. Thus,

$$\mathbf{u} = \mathbf{Proj}_U(\mathbf{v}) = \mathbf{Proj}_{\mathbf{u}_1}(\mathbf{v}) + \mathbf{Proj}_{\mathbf{u}_2}(\mathbf{v}) = (-1/2)\mathbf{u}_1 + 2\mathbf{u}_2 = \frac{-1}{2}\begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} + 2\begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3/2 \\ 5/2 \\ 2 \\ 4 \end{bmatrix}.$$

We also verify numerically that the difference $\mathbf{v}' = \mathbf{v} - \mathbf{Proj}_U(\mathbf{v}) = \mathbf{v} - \mathbf{u}$ is orthogonal to everything in U, or in other words is orthogonal to $\mathbf{u}_1$ and $\mathbf{u}_2$, as it must be if we have not made a mistake. We compute

$$\mathbf{v}' = \mathbf{v} - \mathbf{u} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} - \begin{bmatrix} 3/2 \\ 5/2 \\ 2 \\ 4 \end{bmatrix} = \begin{bmatrix} -1/2 \\ -1/2 \\ 1 \\ 0 \end{bmatrix},$$

so $\mathbf{v}' \cdot \mathbf{u}_1 = -1/2 - (-1/2) = 0$ and $\mathbf{v}' \cdot \mathbf{u}_2 = -1/2 - 1/2 + 1 + 0 = 0$ as expected. ■

Here is a reformulation of Theorem 6.2.1, expressing the insight of an orthogonality characterization of displacement from the nearest point.

Theorem 6.2.4 (Orthogonal Projection Theorem, version II). If V is a linear subspace of $\mathbf{R}^n$ then every vector $\mathbf{x} \in \mathbf{R}^n$ can be *uniquely* expressed as a sum

$$\mathbf{x} = \mathbf{v} + \mathbf{v}'$$

with $\mathbf{v} \in V$ and $\mathbf{v}'$ orthogonal to everything in V. Explicitly, $\mathbf{v} = \mathbf{Proj}_V(\mathbf{x})$ and $\mathbf{v}' = \mathbf{x} - \mathbf{Proj}_V(\mathbf{x})$.

The preceding reformulation is an immediate consequence of the assertion in Theorem 6.2.1 that $\mathbf{Proj}_V(\mathbf{x})$ is the *only* vector in V whose difference from $\mathbf{x}$ is perpendicular to *everything* in V.

Remark 6.2.5. If you are convinced by the discussion of Figure 6.2.1 that the point $\mathbf{v} \in V$ nearest to $\mathbf{x} \in \mathbf{R}^n$ should make $\mathbf{x} - \mathbf{v}$ perpendicular to everything in V, so in effect you find Theorem 6.2.4 to be plausible, let's now see that the formula (6.2.1) then is quite reasonable.

Since the $\mathbf{v}_i$'s span V, the point $\mathbf{v} \in V$ closest to $\mathbf{x}$ can be written in the form $\mathbf{v} = \sum_{i-1}^{k} c_i \mathbf{v}_i$ for some unknown coefficients c_i. We are going to see that the perpendicularity of $\mathbf{x} - \mathbf{v}$ to everything in V forces $c_i = (\mathbf{x} \cdot \mathbf{v}_i)/(\mathbf{v}_i \cdot \mathbf{v}_i)$ for every i. But then $c_i \mathbf{v}_i$ is exactly the formula for $\mathbf{Proj}_{\mathbf{v}_i}(\mathbf{x})$ in Proposition 6.1.1, so we would obtain $\mathbf{v} = \sum_{i=1}^{k} c_i \mathbf{v}_i = \sum_{i=1}^{k} \mathbf{Proj}_{\mathbf{v}_i}(\mathbf{x})$ as asserted in (6.2.1).

How can we show that the coefficients c_i are really given by the ratios $(\mathbf{x} \cdot \mathbf{v}_i)/(\mathbf{v}_i \cdot \mathbf{v}_i)$? For this we have to do some algebra (rather than geometry): since $\mathbf{x} - \mathbf{v}$ is perpendicular to everything in V, it is in particular perpendicular to every $\mathbf{v}_j$, so

$$0 = (\mathbf{x} - \mathbf{v}) \cdot \mathbf{v}_j = \mathbf{x} \cdot \mathbf{v}_j - \mathbf{v} \cdot \mathbf{v}_j$$

for every j. This says $\mathbf{x} \cdot \mathbf{v}_j = \mathbf{v} \cdot \mathbf{v}_j$ for every j. But $\mathbf{v} = \sum_{i=1}^{k} c_i \mathbf{v}_i$ with some unknown c_i's, so

$$\mathbf{v} \cdot \mathbf{v}_j = \sum_{i=1}^{k} (c_i \mathbf{v}_i) \cdot \mathbf{v}_j = \sum_{i=1}^{k} c_i (\mathbf{v}_i \cdot \mathbf{v}_j),$$

and the terms for $i \neq j$ all vanish since $\mathbf{v}_i \cdot \mathbf{v}_j = 0$ whenever $i \neq j$ (as $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ is an *orthogonal* basis of V!). In other words, $\mathbf{v} \cdot \mathbf{v}_j = c_j (\mathbf{v}_j \cdot \mathbf{v}_j)$ for every j. But we have seen that $\mathbf{x} \cdot \mathbf{v}_j = \mathbf{v} \cdot \mathbf{v}_j$, so

$$\mathbf{x} \cdot \mathbf{v}_j = c_j (\mathbf{v}_j \cdot \mathbf{v}_j)$$

for every j. We can divide by $\mathbf{v}_j \cdot \mathbf{v}_j$ since this is nonzero (it is equal to $\|\mathbf{v}_j\|^2 > 0$, as $\mathbf{v}_j \neq \mathbf{0}$), so we thereby obtain the formula $c_j = (\mathbf{x} \cdot \mathbf{v}_j)/(\mathbf{v}_j \cdot \mathbf{v}_j)$ for every j, as desired.

Remark 6.2.6. In this section, we have solved the problem of computing the point in a linear subspace of $\mathbf{R}^n$ closest to a chosen point $\mathbf{x} \in \mathbf{R}^n$ by using an orthogonal basis for the subspace. The task of actually *finding* an orthogonal basis for a linear subspace of $\mathbf{R}^n$ has not yet been addressed. We'll take up a special case in Section 7.1 and will solve it general in Chapter 19 using a geometric technique called the Gram–Schmidt process. An alternative algebraic method (using techniques in "matrix algebra") will be given in Chapter 20 (see Theorem 20.6.3).

6.3. Determination of closest point. In this section, we give a proof of Theorem 6.2.1, partly motivated by Figure 6.2.1 that illustrates it (as a guide for expecting certain dot products to be equal to 0, which we shall verify by calculation). The case $V = \{\mathbf{0}\}$ is easy by unraveling definitions, so we may and do assume V is nonzero.

Choose an orthogonal basis $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ of V and *define* the notation $\mathbf{Proj}_V(\mathbf{x})$ by the formula (6.2.1) in terms of that chosen orthogonal basis. We proceed in several steps to show that this point satisfies all of the desired properties. (Strictly speaking, this proof is assuming the existence of an orthogonal basis of V. We will establish the existence of such bases for nonzero linear subspaces in general in Chapter 19, not relying on the present considerations. So at that time our proof of Theorem 6.2.1 will be complete.)

Step 1. Show $\mathbf{x} - \mathbf{Proj}_V(\mathbf{x})$ is perpendicular to every $\mathbf{v}_i$.

We will show each dot product $(\mathbf{x} - \mathbf{Proj}_V(\mathbf{x})) \cdot \mathbf{v}_i$ vanishes. The dot product $\mathbf{v}_i \cdot \mathbf{v}_j$ vanishes for $j \neq i$ because the $\mathbf{v}_i$'s are orthogonal to each other by assumption. Since $\mathbf{Proj}_{\mathbf{v}_j}(\mathbf{x})$ is a scalar multiple of $\mathbf{v}_j$, it follows that $\mathbf{Proj}_{\mathbf{v}_j}(\mathbf{x})\cdot\mathbf{v}_i = 0$ for each $j \neq i$ as well. Therefore

$$(\mathbf{x}-\mathbf{Proj}_V(\mathbf{x}))\cdot\mathbf{v}_i = (\mathbf{x}-\mathbf{Proj}_{\mathbf{v}_i}\,\mathbf{x})\cdot\mathbf{v}_i + \text{ other terms which vanish.}$$

To check that $(\mathbf{x}-\mathbf{Proj}_{\mathbf{v}_i}\,\mathbf{x})\cdot\mathbf{v}_i = 0$ too, we use the formula $\mathbf{Proj}_{\mathbf{v}_i}\,\mathbf{x} = \left(\dfrac{\mathbf{x}\cdot\mathbf{v}_i}{\mathbf{v}_i\cdot\mathbf{v}_i}\right)\mathbf{v}_i$ to get

$$(\mathbf{x}-\mathbf{Proj}_{\mathbf{v}_i}\,\mathbf{x})\cdot\mathbf{v}_i = \mathbf{x}\cdot\mathbf{v}_i - (\mathbf{Proj}_{\mathbf{v}_i}\,\mathbf{x})\cdot\mathbf{v}_i = \mathbf{x}\cdot\mathbf{v}_i - \left(\left(\frac{\mathbf{x}\cdot\mathbf{v}_i}{\mathbf{v}_i\cdot\mathbf{v}_i}\right)\mathbf{v}_i\right)\cdot\mathbf{v}_i. \qquad (6.3.1)$$

But $(c\mathbf{v}_i)\cdot\mathbf{v}_i = c(\mathbf{v}_i\cdot\mathbf{v}_i)$ for any scalar c (see Theorem 2.2.1(iii)), so the term being subtracted on the right side of (6.3.1) equals $\left(\dfrac{\mathbf{x}\cdot\mathbf{v}_i}{\mathbf{v}_i\cdot\mathbf{v}_i}\right)\mathbf{v}_i\cdot\mathbf{v}_i = \mathbf{x}\cdot\mathbf{v}_i$ by cancellation of the denominator. Hence, the difference at the end of (6.3.1) vanishes.

Step 2. $\mathbf{x}-\mathbf{Proj}_V(\mathbf{x})$ is perpendicular to every vector in V.

We must show the dot product of $\mathbf{x}-\mathbf{Proj}_V(\mathbf{x})$ with every vector in V is zero. Every vector $\mathbf{v}\in V$ is a linear combination $\mathbf{v} = \sum_{i=1}^k c_i\mathbf{v}_i$ of the vectors $\mathbf{v}_1,\dots,\mathbf{v}_k$ (since the $\mathbf{v}_i$'s span V, being a basis of V), and we can move dot products through linear combinations (see Theorem 2.2.1). Thus,

$$(\mathbf{x}-\mathbf{Proj}_V(\mathbf{x}))\cdot\mathbf{v} = (\mathbf{x}-\mathbf{Proj}_V(\mathbf{x}))\cdot\sum_{i=1}^k c_i\mathbf{v}_i = \sum_{i=1}^k c_i\left((\mathbf{x}-\mathbf{Proj}_V(\mathbf{x}))\cdot\mathbf{v}_i\right) = \sum_{i=1}^k c_i(0) = 0,$$

using Step 1 at the end.

Step 3. $\mathbf{Proj}_V(\mathbf{x})$ is the closest point on V to $\mathbf{x}$, and the only point attaining the minimal distance.

Any vector $\mathbf{v}\in V$ can be written as $\mathbf{v} = \mathbf{Proj}_V(\mathbf{x})+\mathbf{y}$ where $\mathbf{y} = \mathbf{v}-\mathbf{Proj}_V(\mathbf{x})\in V$. Let's see how far $\mathbf{v}$ is from $\mathbf{x}$ in terms of $\mathbf{y}$:

$$\|\mathbf{x}-\mathbf{v}\| = \|\mathbf{x}-(\mathbf{Proj}_V(\mathbf{x})+\mathbf{y})\| = \|(\mathbf{x}-\mathbf{Proj}_V(\mathbf{x}))-\mathbf{y}\| = \sqrt{\|\mathbf{x}-\mathbf{Proj}_V(\mathbf{x})\|^2+\|-\mathbf{y}\|^2},$$

where the last equality is the Pythagorean Theorem (Theorem 2.3.1), applicable because $\mathbf{x}-\mathbf{Proj}_V(\mathbf{x})$ is perpendicular to $-\mathbf{y}\in V$ by Step 2! If $\mathbf{y}$ is *nonzero* (which is to say $\mathbf{v}\neq\mathbf{Proj}_V(\mathbf{x})$), so $\|-\mathbf{y}\|^2 > 0$, then the right side is larger than $\|\mathbf{x}-\mathbf{Proj}_V(\mathbf{x})\|$. Hence, the point in V minimizing the distance to $\mathbf{x}$ is $\mathbf{v} = \mathbf{Proj}_V(\mathbf{x})$ and it is the only point attaining that minimal distance.

Step 4. The only point $\mathbf{v}\in V$ for which $\mathbf{x}-\mathbf{v}$ is perpendicular to everything in V is $\mathbf{Proj}_V(\mathbf{x})$.

We have seen that $\mathbf{x}-\mathbf{Proj}_V(\mathbf{x})$ is perpendicular to everything in V, and must show that if $\mathbf{v}\in V$ makes $\mathbf{x}-\mathbf{v}$ orthogonal to everything in V then $\mathbf{v} = \mathbf{Proj}_V(\mathbf{x})$. The assumption for a point $\mathbf{v}\in V$ that $\mathbf{x}-\mathbf{v}$ is perpendicular to everything in V is exactly what was used in Remark 6.2.5 to deduce that $\mathbf{v} = \sum_{i=1}^k \mathbf{Proj}_{\mathbf{v}_i}(\mathbf{x})$, and this sum is what we defined the notation $\mathbf{Proj}_V(\mathbf{x})$ to mean in this proof.

Chapter 6 highlights (links to highlights in and next chapters)

Notation	**Meaning**	**Location in text**
$\mathbf{Proj}_{\mathbf{w}}(\mathbf{x})$	for nonzero $\mathbf{w} \in \mathbf{R}^n$ and any $\mathbf{x} \in \mathbf{R}^n$, the unique point in $\mathrm{span}(\mathbf{w})$ closest to $\mathbf{x}$	Figure 6.1.1, Proposition 6.1.1
$\mathbf{Proj}_V(\mathbf{x})$	for a linear subspace V in $\mathbf{R}^n$ and any $\mathbf{x} \in \mathbf{R}^n$, the unique point in V closest to $\mathbf{x}$	Theorem 6.2.1

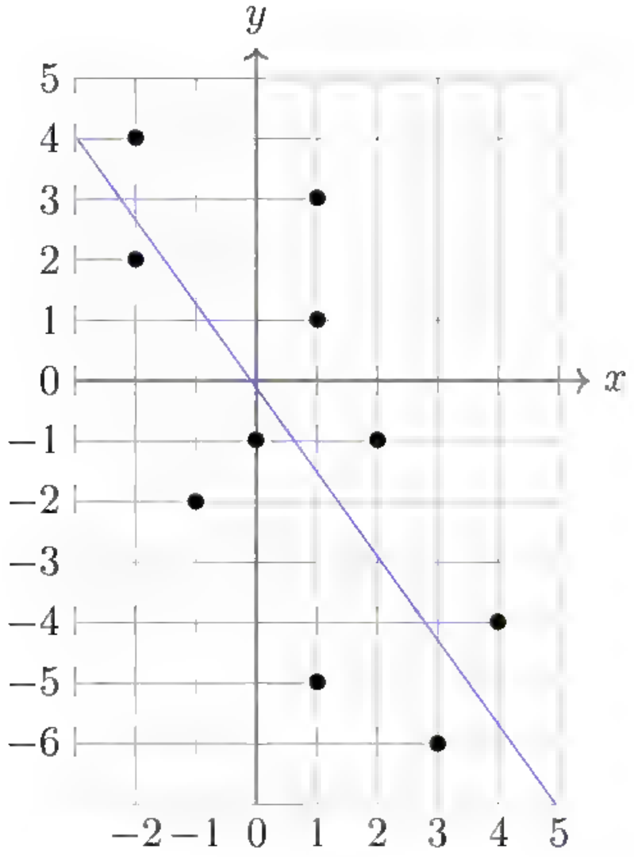
Concept	**Meaning**	**Location in text**
projection onto a nonzero n-vector $\mathbf{w}$	for any $\mathbf{x} \in \mathbf{R}^n$, the unique point in $\mathrm{span}(\mathbf{w})$ closest to $\mathbf{x}$	Figure 6.1.1, Proposition 6.1.1
projection onto a linear subspace V in $\mathbf{R}^n$	for any $\mathbf{x} \in \mathbf{R}^n$, the unique point in V closest to $\mathbf{x}$	Theorem 6.2.1

Result	**Meaning**	**Location in text**
$\mathbf{Proj}_{\mathbf{w}}$ formula	for any $\mathbf{x} \in \mathbf{R}^n$, $\mathbf{Proj}_{\mathbf{w}}(\mathbf{x}) = \left(\frac{\mathbf{x} \cdot \mathbf{w}}{\mathbf{w} \cdot \mathbf{w}}\right) \mathbf{w}$	(6.1.2)
$\mathbf{Proj}_{\mathbf{w}}$ interacts well with linear combinations	applying $\mathbf{Proj}_{\mathbf{w}}$ to a linear combination yields the corresponding linear combination of $\mathbf{Proj}_{\mathbf{w}}$'s: for any n-vectors $\mathbf{x}_1, \mathbf{x}_2$ and scalars c_1, c_2, we have $\mathbf{Proj}_{\mathbf{w}}(c_1\mathbf{x}_1 + c_2\mathbf{x}_2) = c_1\,\mathbf{Proj}_{\mathbf{w}}(\mathbf{x}_1) + c_2\,\mathbf{Proj}_{\mathbf{w}}(\mathbf{x}_2)$	(6.1.3)
Orthogonal Projection Theorem	for $\mathbf{x} \in \mathbf{R}^n$ and a linear subspace V in $\mathbf{R}^n$, the unique $\mathbf{v} \in V$ making $\lVert\mathbf{x} - \mathbf{v}\rVert$ minimal also makes $\mathbf{x} - \mathbf{v}$ orthogonal to everything in V; we denote this $\mathbf{v}$ as $\mathbf{Proj}_V(\mathbf{x})$, and explicitly if $\mathbf{v}_1, \dots, \mathbf{v}_k$ is an orthogonal basis of V then $\mathbf{Proj}_V(\mathbf{x}) = \mathbf{Proj}_{\mathbf{v}_1}(\mathbf{x}) + \cdots + \mathbf{Proj}_{\mathbf{v}_k}(\mathbf{x})$	Theorem 6.2.1, Figure 6.2.1
every n-vector $\mathbf{x}$ is uniquely sum of a vector in V and a vector orthogonal to everything in V	for every linear subspace V in $\mathbf{R}^n$ and $\mathbf{x} \in \mathbf{R}^n$, we can write in exactly one way $\mathbf{x} = \mathbf{v} + \mathbf{v}'$ with $\mathbf{v} \in V$ and $\mathbf{v}'$ orthogonal to everything in V (and in fact $\mathbf{v} = \mathbf{Proj}_V(\mathbf{x})$)	Theorem 6.2.4

Skill	**Location in text**
compute $\mathbf{Proj}_{\mathbf{w}}(\mathbf{x})$ for $\mathbf{x} \in \mathbf{R}^n$ and nonzero $\mathbf{w} \in \mathbf{R}^n$	Examples 6.1.2 and 6.1.4
compute $\mathbf{Proj}_{\mathbf{w}}(\mathbf{x})$ as linear combination of $\mathbf{Proj}_{\mathbf{w}}(\mathbf{e}_1), \dots, \mathbf{Proj}_{\mathbf{w}}(\mathbf{e}_n)$	(6.1.4), Figure 6.1.5
compute $\mathbf{Proj}_V(\mathbf{x})$ using an orthogonal basis of V	(6.2.1), Example 6.2.3

7. Applications of projections in $\mathbf{R}^n$: orthogonal bases of planes and linear regression

Linear regression refers to the problem of finding a function $f(x) = mx + b$ which best fits a collection of given data points (x_i, y_i) (as well as variants of this problem, sometimes using curves other than a line).



The extent to which this line fits the given data is measured by a number called the *correlation coefficient*, which was introduced in Section 2.4. One of the themes of this chapter is using linear algebra in $\mathbf{R}^n$ for big n to optimally fit a *line* to n data points in $\mathbf{R}^2$. The technique we employ will involve computing projection to a 2-dimensional linear subspace of $\mathbf{R}^n$, so as a preliminary step we will develop a technique for computing *orthogonal* bases of such subspaces (since we need an orthogonal basis of a linear subspace in order to compute projection to that subspace via the formula in Theorem 6.2.1).

We will return to lines of best fit in Section 20.6 to understand them in another way via matrix algebra that is also applicable to approximating data in $\mathbf{R}^2$ with a curve $y = f(x)$ for higher-degree polynomials $f(x)$.

By the end of this chapter, you should be able to do the following:

- find an orthogonal basis for a 2-dimensional subspace V (a "plane") in $\mathbf{R}^n$;
- set up the problem of a "best fit line" to n data points in $\mathbf{R}^2$ using the language of linear algebra and 2-dimensional subspaces of $\mathbf{R}^n$;
- compute that line, using projections and orthogonal bases of planes in $\mathbf{R}^n$.

The techniques of this chapter are just the tip of the iceberg on applying ideas of high-dimensional linear algebra to the study of data. For example, at the end of the book (in Chapter 27) we will discuss how more advanced results in high-dimensional linear algebra underlie *principal component analysis* (PCA) that approximates data in $\mathbf{R}^n$ for gigantic n using a "low-dimensional" linear subspace (with dimension bigger than 1, so going beyond lines as in this chapter); this is the most important algorithm in contemporary data science, pervading all modern quantitative fields.

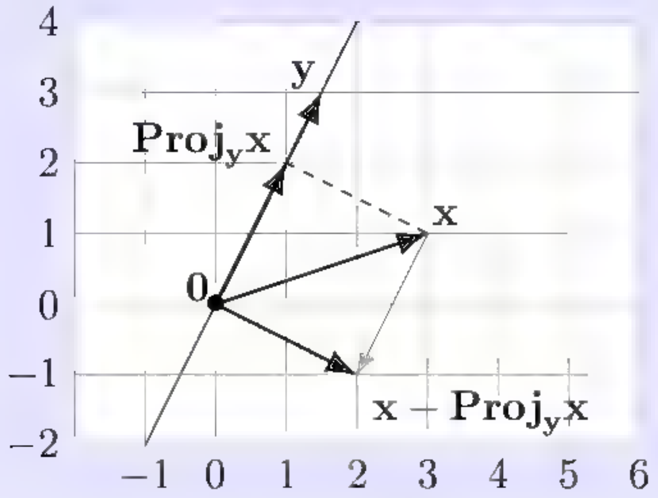
7.1. Finding an orthogonal basis: special case. Our formula for projection to a linear subspace of $\mathbf{R}^n$ in Theorem 6.2.1 uses an orthogonal basis of the linear subspace. But we can also go the other way:

use projections to find an orthogonal basis for a linear subspace! This might sound circular, but it is not: the method is to interweave the processes of computing projections and finding orthogonal bases by using *lower-dimensional* subspaces (so we build up to the desired calculation in a step-by-step manner through "smaller" situations). This will be carried out in general in Chapter 19, and now we explain how to do this for a 2-dimensional subspace of $\mathbf{R}^n$ by using projection onto lines via dot products (Proposition 6.1.1).

Theorem 7.1.1. Suppose $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$ are nonzero, and not scalar multiples of each other. The vectors

$$\mathbf{y} \quad \text{and} \quad \mathbf{x}' = \mathbf{x} - \mathbf{Proj}_{\mathbf{y}}\,\mathbf{x} \tag{7.1.1}$$

constitute an orthogonal basis of $\mathrm{span}(\mathbf{x}, \mathbf{y})$. (In particular, $\mathrm{span}(\mathbf{x}, \mathbf{y})$ is 2-dimensional.)



The setup is symmetric in $\mathbf{x}$ and $\mathbf{y}$, so $\{\mathbf{x}, \mathbf{y}' = \mathbf{y} - \mathbf{Proj}_{\mathbf{x}}\,\mathbf{y}\}$ is also an orthogonal basis of $\mathrm{span}(\mathbf{x}, \mathbf{y})$.

A proof of Theorem 7.1.1 is given at the start of Section 7.5 for those who are interested. It must be emphasized that although the picture in Theorem 7.1.1 is in $\mathbf{R}^2$ to help us to remember what it says, the result as stated really is building an orthogonal basis from a given basis of any 2-dimensional subspace of $\mathbf{R}^n$ for any n. The applicability in $\mathbf{R}^n$ for any n will be essential for computing the line of best fit for n data points in $\mathbf{R}^2$ in Section 7.3.

An important application of Theorem 7.1.1, also relevant to lines of best fit later on, is to compute projections to 2-dimensional linear subspaces. The following example illustrates how this can arise.

Example 7.1.2. Consider the plane V in $\mathbf{R}^3$ through $\mathbf{0}$ spanned by the vectors

$$\mathbf{v} = \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} 0 \\ 3 \\ 4 \end{bmatrix}.$$

Imagine that this plane is a metal sheet on which an electric charge is uniformly distributed. An iron particle placed at the point $\mathbf{p} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ would then be attracted to the metal sheet, and by the symmetry of the situation this particle would move straight towards the point on the plane closest to the initial position of the particle. What is that point?

In other words, we seek to compute the projection $\mathbf{Proj}_V(\mathbf{p})$. To compute this, we first seek an orthogonal basis for the plane V. By Theorem 7.1.1, such an orthogonal basis is given by $\mathbf{w}$ and $\mathbf{v}' = \mathbf{v} - \mathbf{Proj}_{\mathbf{w}}(\mathbf{v})$. We first compute $\mathbf{Proj}_{\mathbf{w}}(\mathbf{v})$. This is given by

$$\mathbf{Proj}_{\mathbf{w}}(\mathbf{v}) = \left(\frac{\mathbf{v} \cdot \mathbf{w}}{\mathbf{w} \cdot \mathbf{w}}\right) \mathbf{w} = \frac{3}{25} \begin{bmatrix} 0 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 9/25 \\ 12/25 \end{bmatrix}.$$

Thus $\mathbf{v}' = \mathbf{v} - \begin{bmatrix} 0 \\ 9/25 \\ 12/25 \end{bmatrix} = \begin{bmatrix} 2 \\ 16/25 \\ -12/25 \end{bmatrix}$. (As a safety check, $\mathbf{w}$ and $\mathbf{v}'$ are indeed orthogonal.)

The vector $\mathbf{v}'$ is a bit ugly due to the fractions, and for the purposes of having an orthogonal basis it is harmless to replace it with a nonzero scalar multiple, such as

$$\mathbf{v}'' = 25\mathbf{v}' = \begin{bmatrix} 50 \\ 16 \\ -12 \end{bmatrix}.$$

Since $\{\mathbf{w}, \mathbf{v}''\}$ is an orthogonal basis of the plane V, we have

$$\mathbf{Proj}_V(\mathbf{p}) = \mathbf{Proj}_V \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \mathbf{Proj}_{\mathbf{w}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \mathbf{Proj}_{\mathbf{v}''} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \mathbf{Proj}_{\mathbf{w}}(\mathbf{p}) + \mathbf{Proj}_{\mathbf{v}''}(\mathbf{p}).$$

To compute these projections, we first work out some relevant dot products:

$$\mathbf{w} \cdot \mathbf{w} = 25, \quad \mathbf{v}'' \cdot \mathbf{v}'' = 2900, \quad \mathbf{p} \cdot \mathbf{w} = 7, \quad \mathbf{p} \cdot \mathbf{v}'' = 54.$$

Hence

$$\mathbf{Proj}_{\mathbf{w}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \mathbf{Proj}_{\mathbf{w}}(\mathbf{p}) = \left(\frac{\mathbf{p} \cdot \mathbf{w}}{\mathbf{w} \cdot \mathbf{w}}\right) \mathbf{w} = \frac{7}{25} \begin{bmatrix} 0 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 21/25 \\ 28/25 \end{bmatrix},$$

$$\mathbf{Proj}_{\mathbf{v}''} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \mathbf{Proj}_{\mathbf{v}''}(\mathbf{p}) = \left(\frac{\mathbf{p} \cdot \mathbf{v}''}{\mathbf{v}'' \cdot \mathbf{v}''}\right) \mathbf{v}'' = \frac{54}{2900} \begin{bmatrix} 50 \\ 16 \\ -12 \end{bmatrix} = \begin{bmatrix} 27/29 \\ 216/725 \\ -162/725 \end{bmatrix}.$$

(we will **never** ask you to work directly with such complicated fractions; examples in the homework and exams will always involve just integers or fractions with small denominators).

Thus, the place on the metal sheet that the particle ends up at is

$$\mathbf{Proj}_V \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \mathbf{Proj}_{\mathbf{w}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \mathbf{Proj}_{\mathbf{v}''} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 21/25 \\ 28/25 \end{bmatrix} + \begin{bmatrix} 27/29 \\ 216/725 \\ -162/725 \end{bmatrix} = \begin{bmatrix} 27/29 \\ 33/29 \\ 26/29 \end{bmatrix} \approx \begin{bmatrix} 0.931 \\ 1.138 \\ 0.897 \end{bmatrix}.$$

■

7.2. Some motivating examples of linear regression. We now turn our attention to the topic of linear regression, which will be understood via Theorem 7.1.1. First, here are some examples illustrating the ubiquity of linear regression.

Example 7.2.1. (The original example) On January 1, 1801, astronomers found what they thought was a new planet (it was later decided it was an asteroid, Ceres). They "lost" it in February of the same year, because its orbit was rendered invisible by the position of the sun. (This was a big deal, because Ceres was thought to be a new planet; it only got downgraded later.) The 24-year old mathematician C.F. Gauss[11] used a version of what came to be called the "method of least squares" to predict the orbit of Ceres, and it was re-located in December 1801 according to Gauss' computations (thereby making him famous far beyond the community of mathematicians). ■

---

[11] Carl Friedrich Gauss (1777-1855) was the greatest mathematician of the 19th century, in terms of creativity and influence. He taught himself arithmetic before he knew how to speak, memorized an entire table of logarithms to speed up his mental calculations, proved spectacular new theorems during his teenage years (solving problems that had stumped the ancient Greeks and Euler), made revolutionary discoveries in numerous fields such as number theory, differential geometry, algebra, probability, and physics, and spoke at least 7 languages fluently. He also invested wisely, leaving an estate worth 200 times his annual salary.

Example 7.2.2. Linear regression is used to estimate how a change in one variable is related to a change in another (when data supports that the variables are related in an approximately "linear" way). For instance:

- "A one percentile increase in end-of-kindergarten test scores is associated with a \$132 increase in wage earnings at age 27." [**CFHSSY**, Sec. I]
- "An extra \$100,000 in campaign spending garners a challenger [an extra] 0.3% of the vote." [**Le**, p. 780]
- "An extra inch [of height] is worth almost \$800 a year in elevated earnings." [**Pi**]

■

Example 7.2.3. My fancy watch tells me how many calories I've burned in a given day. It doesn't measure this; rather it (presumably) uses one of several formulas to estimate this from height, weight, age and other variables. These formulas are based on experimental studies that use linear regression.

For example, the first such formula was the Harris–Benedict equation (which estimated the basal metabolic rate (in kcal/day) to be $655 + 9.56\,M + 1.85\,H - 4.68\,A$ for women and $66.47 + 13.75\,M + 5.0\,H - 6.755\,A$ for men, where M is mass (in kg), H is height (in cm), and A is age (in years)). This was based on a 1918 study [**HB**] of 103 women, 136 men, and 94 newborn infants. It was subsequently revised in 1984 to account for lifestyle changes since 1918. (Section 3.1 of the Wikipedia page on "basal metabolic rate" gives several such formulas.) ■

Beware that correlation (or good approximation by a best-fit line) does not imply causation! Failure to keep this in mind is a common fallacy when using linear regression. There is no reason that effects related to one of the two variables in a linear regression are "caused" by the other variable. For a truly hilarious collection of spurious correlations, read the book [**Vig**].

7.3. Fitting a function to data.

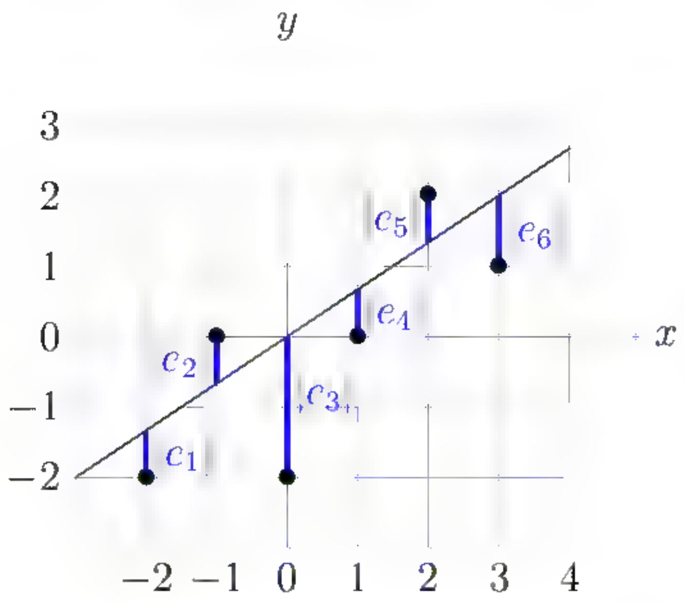
The plain vanilla version of our problem is as follows:

find the function $f(x) = mx + b$ that "best fits" n data points $(x_1, y_1), \ldots, (x_n, y_n)$.

What does "best fit" mean? Informally, we want $f(x_i)$ to be as close as possible to y_i for all i. The error

$$\text{error}_i = y_i - (mx_i + b)$$

measures in absolute value how close the line $y = mx + b$ is vertically to (x_i, y_i); see Figure 7.3.1.



Suppose the line is given by the equation $y = mx + b$.
Suppose the ith data point is denoted (x_i, y_i).
The ith error is given by $\text{error}_i = e_i = y_i - (mx_i + b)$.
These errors are shown as blue line segments in the figure.

FIGURE 7.3.1. Errors are measured vertically, **not** the actual distance from data points to the line! The way we *use* the best-fit line explains why vertical distance is the right notion of "error".

To be a “good fit” means to choose (m, b) so that the errors are collectively small. There are many ways to specify what “collectively small” means. The meaning in the *least squares method* is this:

> choose (m, b) to minimize the sum of the squares of the errors; i.e., choose (m, b) to minimize
> $$\sum_{i=1}^{n}(y_i - (mx_i + b))^2.$$

Remark 7.3.1. Why use the *sum of squares* of the errors? The errors themselves might be positive and might be negative; we want to penalize a large negative error as well as a large positive error, so squaring errors removes the sign. Ultimately we *define* whatever measure of best fit we will use (one cannot “derive” or “prove” a definition); the real test of a definition is its utility in applications. From that perspective “sum of squares of errors” is a good definition: experience shows that the sum-of-squares idea works really well across a broad range of applications, and it has an array of convenient mathematical properties (e.g., below we’ll give it a nice geometric meaning in terms of distance minimization in $\mathbf{R}^n$, and it relates well to correlation coefficients as recorded in (7.3.5)).

But sometimes other ways to *define* the “total error” are indeed more appropriate, such as summing the absolute values of the errors (used in computational statistics, geophysics, and the important signal processing algorithm called “compressed sensing”). The absolute value function is inconvenient for our purposes; e.g., from a calculus viewpoint, $|x|$ has the defect relative to x^2 that it is not differentiable at $x = 0$. Always remember that we *choose* how to define “total error” for any particular application, and experience determines the appropriateness of that choice; mathematics is a creation of the human mind.

In the rather special case that all n data points (x_i, y_i) lie on the same vertical line, which is to say all x_i’s share the same value, that vertical line obviously does the job. Thus, we assume the data are *not* on a single vertical line. We can write our task more succinctly using vector language, as follows.

Put the data of all x-values into a single n-vector, and the data of all y-values into a single n-vector:

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

Also, define $\mathbf{1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbf{R}^n$ to be the vector with all entries equal to 1 (analogous to $\mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbf{R}^n$), so

$$m\mathbf{X} + b\mathbf{1} = \begin{bmatrix} mx_1 \\ mx_2 \\ \vdots \\ mx_n \end{bmatrix} + \begin{bmatrix} b \\ b \\ \vdots \\ b \end{bmatrix} = \begin{bmatrix} mx_1 + b \\ mx_2 + b \\ \vdots \\ mx_n + b \end{bmatrix}$$

and hence

$$\mathbf{Y} - (m\mathbf{X} + b\mathbf{1}) = \begin{bmatrix} y_1 - (mx_1 + b) \\ y_2 - (mx_2 + b) \\ \vdots \\ y_n - (mx_n + b) \end{bmatrix} = \text{“vector of errors”}.$$

Thus, since $\sum_{i=1}^{n} v_i^2 = \|\mathbf{v}\|^2$ for any $\mathbf{v} \in \mathbf{R}^n$ (by *definition* of $\|\mathbf{v}\|$!), the sum of the squares of the errors is

$$\sum_{i=1}^{n}(y_i - (mx_i + b))^2 = \|\mathbf{Y} - (m\mathbf{X} + b\mathbf{1})\|^2;$$

this is the squared length of $\mathbf{Y} - (m\mathbf{X} + b\mathbf{1})$ due to how "length" for n-vectors is *defined*. So we seek m and b that minimize the squared length of $\mathbf{Y} - (m\mathbf{X} + b\mathbf{1})$, which is the *same* as minimizing the length of that difference. We are now going to use the language of linear algebra to interpret this task in a visual way which opens the door to applying earlier geometric results (with projections).

The length $\|\mathbf{Y} - (m\mathbf{X} + b\mathbf{1})\|$ is the distance from $\mathbf{Y}$ to $m\mathbf{X} + b\mathbf{1}$ since "distance" between any n-vectors $\mathbf{v}$ and $\mathbf{w}$ is $\|\mathbf{v} - \mathbf{w}\|$ by *definition*. As m and b vary, the vectors of the form $m\mathbf{X} + b\mathbf{1}$ are exactly the vectors in $\mathrm{span}(\mathbf{X}, \mathbf{1})$, due to the *definition* of "span". Hence, the least-squares minimization problem for n data points is *equivalent* to the following geometric problem:

$$\text{find the vector in } \mathrm{span}(\mathbf{X}, \mathbf{1}) \text{ that is closest to the vector } \mathbf{Y} \in \mathbf{R}^n. \tag{7.3.1}$$

Our task is now an instance of finding the point of a linear subspace of $\mathbf{R}^n$ closest to a given n-vector, which we discussed in Chapter 6. Theorem 6.2.1 gives a formula for the answer if we know an *orthogonal* basis for $\mathrm{span}(\mathbf{X}, \mathbf{1})$. We have also seen, in Theorem 7.1.1, how to find an orthogonal basis for a span of two vectors that are not scalar multiples of each other. The vectors $\mathbf{X}$ and $\mathbf{1}$ are not scalar multiples of each other because the hypothesis that the n data points do not lie in a common vertical line (i.e., the x_i's are not all equal to each other) says that $\mathbf{X}$ is not a scalar multiple of the nonzero vector $\mathbf{1}$.

By Theorem 7.1.1, an orthogonal basis of $\mathrm{span}(\mathbf{X}, \mathbf{1})$ is given by $\mathbf{1}$ and $\widehat{\mathbf{X}} = \mathbf{X} - \mathbf{Proj_1}\,\mathbf{X}$ with

$$\mathbf{Proj_1}(\mathbf{X}) = \frac{\mathbf{X}\cdot\mathbf{1}}{\mathbf{1}\cdot\mathbf{1}}\mathbf{1} = \frac{\sum_{i=1}^n x_i \cdot 1}{\sum_{i=1}^n 1\cdot 1}\mathbf{1} = \frac{\sum_{i=1}^n x_i}{n}\mathbf{1} = \overline{x}\,\mathbf{1} = \begin{bmatrix} \overline{x} \\ \overline{x} \\ \vdots \\ \overline{x} \end{bmatrix}$$

equal to the n-vector each of whose entries is equal to the average $\overline{x}$ of the x_i's. Hence,

$$\widehat{\mathbf{X}} = \mathbf{X} - \mathbf{Proj_1}(\mathbf{X}) = \begin{bmatrix} x_1 - \overline{x} \\ x_2 - \overline{x} \\ \vdots \\ x_n - \overline{x} \end{bmatrix} \tag{7.3.2}$$

is obtained from $\mathbf{X}$ by subtracting the average $\overline{x}$ from all entries. By applying to this span the formula in (6.2.1) for the nearest point on a linear subspace in terms of an *orthogonal basis*, we obtain that

$$\text{closest vector to } \mathbf{Y} \text{ in } \mathrm{span}(\mathbf{X}, \mathbf{1}) \text{ is } \quad \left(\frac{\mathbf{Y}\cdot\widehat{\mathbf{X}}}{\widehat{\mathbf{X}}\cdot\widehat{\mathbf{X}}}\right)\underbrace{\widehat{\mathbf{X}}}_{\mathbf{X} - \mathbf{Proj_1}\,\mathbf{X}} + \frac{\mathbf{Y}\cdot\mathbf{1}}{\mathbf{1}\cdot\mathbf{1}}\mathbf{1} = \left(\frac{\mathbf{Y}\cdot\widehat{\mathbf{X}}}{\widehat{\mathbf{X}}\cdot\widehat{\mathbf{X}}}\right)\widehat{\mathbf{X}} + \overline{y}\,\mathbf{1} \tag{7.3.3}$$

where $\overline{y} = (1/n)\sum_{i=1}^n y_i$ is the average of the y_i's. The geometry has done its work. In the expression $a\widehat{\mathbf{X}} + \overline{y}\mathbf{1}$ on the right side of (7.3.3), we can expand $\widehat{\mathbf{X}}$ in terms of $\mathbf{X}$ and $\mathbf{1}$ using the definition of $\mathbf{Proj_1}$ and collect terms to rewrite (7.3.3) as a linear combination $m\mathbf{X} + b\mathbf{1}$ of $\mathbf{X}$ and $\mathbf{1}$. Those coefficients m and b are exactly the desired "m" and "b" for the best-fit line! Some examples below carry this out with explicit data to illustrate how it computes m and b systematically; *don't memorize* (7.3.3)!

> Our earlier analysis of closest points to planes in $\mathbf{R}^n$ has now exactly solved the least-squares problem for n data points in $\mathbf{R}^2$ (i.e., finding m and b explicitly) in terms of *computable dot products*. Moreover, the *process* by which we compute m and b has geometric meaning in terms of linear algebra. If formulas for m and b were instead presented as algebra out of thin air then it would have been a mystery where they come from.

Example 7.3.2. For the 5 data points $(-5,-5),(-4,3),(-3,1),(-2,-3),(-1,4)$, let's find the line that best fits it (see Figure 7.3.2). We assemble this information into two 5-vectors:

$$\mathbf{X}=\begin{bmatrix}-5\\-4\\-3\\-2\\-1\end{bmatrix},\mathbf{Y}=\begin{bmatrix}-5\\3\\1\\-3\\4\end{bmatrix}.$$

Next, we find an orthogonal basis for $W=\mathrm{span}(\mathbf{X},\mathbf{1})$. To do this, we first compute the dot products $\mathbf{X}\cdot\mathbf{1}=-15$ and $\mathbf{1}\cdot\mathbf{1}=5$, so $\overline{x}=-15/5=-3$ and hence

$$\widehat{\mathbf{X}}=\mathbf{X}-\mathbf{Proj_1}(\mathbf{X})=\mathbf{X}-\frac{-15}{5}\mathbf{1}=\mathbf{X}+(3)\mathbf{1}=\begin{bmatrix}-2\\-1\\0\\1\\2\end{bmatrix}.$$

Now we can project $\mathbf{Y}$ into $W=\mathrm{span}(\mathbf{X},\mathbf{1})$ using the orthogonal basis $\{\widehat{\mathbf{X}},\mathbf{1}\}$ for W: using the additional dot products

$$\mathbf{Y}\cdot\widehat{\mathbf{X}}=12,\quad \widehat{\mathbf{X}}\cdot\widehat{\mathbf{X}}=10,\quad \mathbf{Y}\cdot\mathbf{1}=0,\quad \mathbf{1}\cdot\mathbf{1}=5,$$

we compute

$$\mathbf{Proj}_W\,\mathbf{Y}=\mathbf{Proj}_{\widehat{\mathbf{X}}}\,\mathbf{Y}+\mathbf{Proj_1}\,\mathbf{Y}=\frac{12}{10}\widehat{\mathbf{X}}+(0)\mathbf{1}=\frac{6}{5}\widehat{\mathbf{X}}.$$

Finally, we substitute the expression $\widehat{\mathbf{X}}=\mathbf{X}+(3)\mathbf{1}$ (as found above) in terms of $\mathbf{X}$ and $\mathbf{1}$ to get

$$\mathbf{Proj}_W\,\mathbf{Y}=\frac{6}{5}(\mathbf{X}+(3)\mathbf{1})=\frac{6}{5}\mathbf{X}+\frac{18}{5}\mathbf{1},$$

so *the best fit line is* $y=(6/5)x+(18/5)$, as shown in Figure 7.3.2. (In Example 7.3.3 we will give a worked example with $\overline{y}\neq 0$, so $\mathbf{Proj_1}\,\mathbf{Y}$ really contributes unlike above where it vanished.)

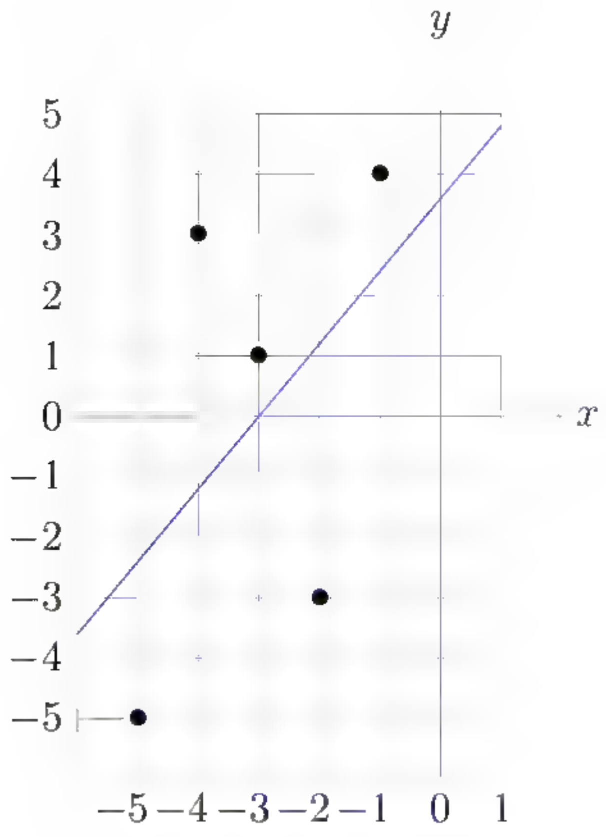


FIGURE 7.3.2. Plot of data and best fit line for Example 7.3.2.

The picture in Figure 7.3.2 indicates a very important issue: we ought to first remove "outlier" points (akin to flawed data). To do this requires methods from statistics beyond the level of this course. ■

Here is the general method. Suppose we are given n data points (x_i, y_i) that do not lie on a common vertical line. To find the best fit line $y = mx + b$, do the following.

(i) Assemble the given data into two n-vectors

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \text{ and } \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

Also let $\mathbf{1}$ be the n-vector all of whose entries are equal to 1 (analogous to $\mathbf{0}$).

(ii) For $W = \text{span}(\mathbf{X}, \mathbf{1})$, we will compute $\mathbf{Proj}_W \mathbf{Y}$ as a linear combination $m\mathbf{X} + b\mathbf{1}$ of $\mathbf{X}$ and $\mathbf{1}$; those coefficients are exactly the coefficients of the best fit line

$$y = mx + b.$$

(iii) To compute $\mathbf{Proj}_W \mathbf{Y}$, use the orthogonal basis $\mathbf{1}$ and $\widehat{\mathbf{X}} = \mathbf{X} - \mathbf{Proj}_{\mathbf{1}} \mathbf{X}$ for W. Explicitly

$$\widehat{\mathbf{X}} = \begin{bmatrix} x_1 - \overline{x} \\ x_2 - \overline{x} \\ \vdots \\ x_n - \overline{x} \end{bmatrix} \text{ and so } \mathbf{Proj}_W(\mathbf{Y}) = \left(\frac{\mathbf{Y} \cdot \widehat{\mathbf{X}}}{\widehat{\mathbf{X}} \cdot \widehat{\mathbf{X}}}\right) \widehat{\mathbf{X}} + \overline{y}\,\mathbf{1} = \left(\frac{\mathbf{Y} \cdot \widehat{\mathbf{X}}}{\widehat{\mathbf{X}} \cdot \widehat{\mathbf{X}}}\right) (\mathbf{X} - \overline{x}\,\mathbf{1}) + \overline{y}\,\mathbf{1},$$

where $\overline{x} = \frac{1}{n}\sum_{i=1}^n x_i$ and $\overline{y} = \frac{1}{n}\sum_{i=1}^n y_i$ are the *averages of the entries of* $\mathbf{X}$ *and* $\mathbf{Y}$.

Whenever $\overline{x} = 0$ we have $\widehat{\mathbf{X}} = \mathbf{X}$. This simplifies things a lot, so we record it here:

$$\text{if } \overline{x} = 0 \text{ then } \mathbf{Proj}_W(\mathbf{Y}) = \left(\frac{\mathbf{Y} \cdot \mathbf{X}}{\mathbf{X} \cdot \mathbf{X}}\right) \mathbf{X} + \overline{y}\,\mathbf{1}, \text{ so } m = \frac{\mathbf{Y} \cdot \mathbf{X}}{\mathbf{X} \cdot \mathbf{X}} \text{ and } b = \overline{y} \text{ whenever } \overline{x} = 0. \quad (7.3.4)$$

WARNING: These formulas for m and b require assuming $\overline{x} = 0$; if $\overline{x} \neq 0$ then they are usually *false*.

Example 7.3.3. Given 4 data points $(-1, 5)$, $(0, 1)$, $(2, -3)$, $(7, -4)$, we'll now use linear algebra in $\mathbf{R}^4$ to find the line of best fit to these points (see Figure 7.3.3 below). We assemble the data into two 4-vectors

$$\mathbf{X} = \begin{bmatrix} -1 \\ 0 \\ 2 \\ 7 \end{bmatrix} \quad \text{and} \quad \mathbf{Y} = \begin{bmatrix} 5 \\ 1 \\ -3 \\ -4 \end{bmatrix}$$

(consisting of the respective x-coordinates and y-coordinates of the points). As above, we also let $\mathbf{1}$ denote the vector $\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$. Note that $\mathbf{X}$ and $\mathbf{1}$ are not scalar multiples of each other (precisely because the x-coordinates are not all the same; i.e., the data points don't all lie on the same vertical line).

We let $W = \text{span}(\mathbf{X}, \mathbf{1})$, and our goal is to compute the projection $\mathbf{Proj}_W \mathbf{Y}$ of $\mathbf{Y}$ into the subspace W, since then upon writing $\mathbf{Proj}_W \mathbf{Y} = m\mathbf{X} + b\mathbf{1}$ the line of best fit is $y = mx + b$.

To compute $\mathbf{Proj}_W\,\mathbf{Y}$, we use the orthogonal basis $\{\widehat{\mathbf{X}}, \mathbf{1}\}$ of W, where

$$\widehat{\mathbf{X}} = \begin{bmatrix} x_1 - \overline{x} \\ x_2 - \overline{x} \\ x_3 - \overline{x} \\ x_4 - \overline{x} \end{bmatrix} \quad \text{for} \quad \overline{x} = \frac{1}{4}\sum_{i=1}^{4} x_i = \frac{1}{4}(-1+0+2+7) = 2.$$

Thus, $\widehat{\mathbf{X}} = \begin{bmatrix} -3 \\ -2 \\ 0 \\ 5 \end{bmatrix}$. Since $\{\widehat{\mathbf{X}}, \mathbf{1}\}$ is an orthogonal basis for W, we have

$$\begin{aligned} \mathbf{Proj}_W\,\mathbf{Y} &= \mathbf{Proj}_{\widehat{\mathbf{X}}}\,\mathbf{Y} + \mathbf{Proj}_{\mathbf{1}}\,\mathbf{Y} \\ &= \left(\frac{\mathbf{Y}\cdot\widehat{\mathbf{X}}}{\widehat{\mathbf{X}}\cdot\widehat{\mathbf{X}}}\right)\widehat{\mathbf{X}} + \left(\frac{\mathbf{Y}\cdot\mathbf{1}}{\mathbf{1}\cdot\mathbf{1}}\right)\mathbf{1}. \end{aligned}$$

We now compute these dot products: $\mathbf{Y}\cdot\widehat{\mathbf{X}} = -37$, $\widehat{\mathbf{X}}\cdot\widehat{\mathbf{X}} = 38$, $\mathbf{Y}\cdot\mathbf{1} = -1$, $\mathbf{1}\cdot\mathbf{1} = 4$. Plugging in:

$$\begin{aligned} \mathbf{Proj}_W\,\mathbf{Y} = \left(\frac{-37}{38}\right)\widehat{\mathbf{X}} + \left(\frac{-1}{4}\right)\mathbf{1} &= -\frac{37}{38}(\mathbf{X} - \overline{x}\,\mathbf{1}) - \frac{1}{4}\mathbf{1} \\ &= -\frac{37}{38}(\mathbf{X} - (2)\mathbf{1}) - \frac{1}{4}\mathbf{1} \\ &= -\frac{37}{38}\mathbf{X} + \left(\frac{74}{38} - \frac{1}{4}\right)\mathbf{1} \\ &= -\frac{37}{38}\mathbf{X} + \frac{129}{76}\mathbf{1}. \end{aligned}$$

Thus the line of best fit is $y = -(37/38)x + 129/76 \approx -0.9737x + 1.6974$.

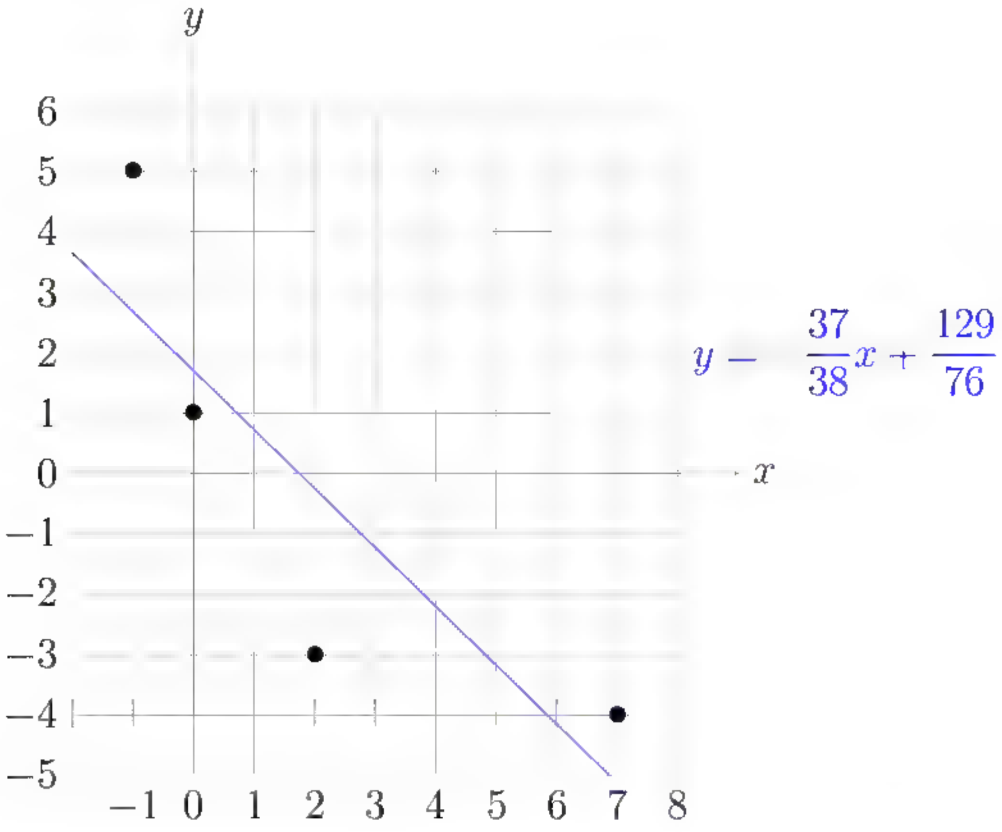


FIGURE 7.3.3. Plot of data and best fit line for Example 7.3.3

Example 7.3.4. For each of the following 5 pairs of data points (seen earlier in Example 2.4.8), the above method can be used to determine the equation $y = mx + b$ of the best-fit line (in the least-squares sense).

(1) $(-2,-4), (-1,-2), (0,0), (1,2), (2,4)$.
(2) $(-2,-5), (-1,3), (0,1), (1,-3), (2,4)$.
(3) $(-2,6), (-1,2), (0,-1), (1,-2), (2,-5)$.
(4) $(-2,4), (-1,-2), (0,-1), (1,3), (2,-4)$.

Before carrying out the calculations, we make some observations. As we noted in Example 2.4.8, the data points in each case have already been adjusted to make $\overline{x}$ and $\overline{y}$ vanish, so (7.3.4) computes m and also that $b = 0$. That is, the best-fit line in each case has the form $y = mx$ for m that we compute from the dot products $\mathbf{Y} \cdot \mathbf{X}$ and $\mathbf{X} \cdot \mathbf{X}$. In all of these cases we have the same $\mathbf{X}$, namely

$$\mathbf{X} = \begin{bmatrix} -2 \\ -1 \\ 0 \\ 1 \\ 2 \end{bmatrix},$$

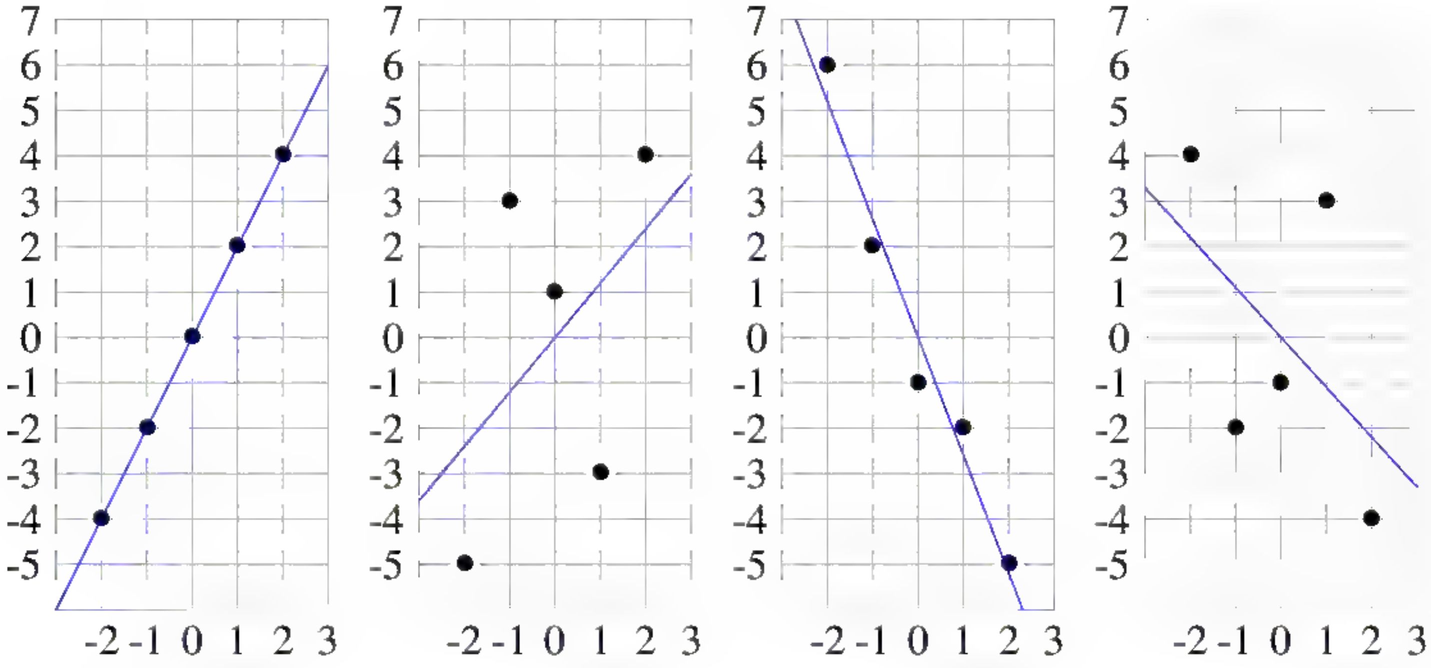
so always $\mathbf{X} \cdot \mathbf{X} = (-2)^2 + (-1)^2 + 0^2 + 1^2 + 2^2 = 4 + 1 + 0 + 1 + 4 = 10$. Hence, in each case the best-fit line is $y = ((\mathbf{Y} \cdot \mathbf{X})/10)x$.

Let's now compute the slope explicitly in each of the four cases. Writing $\mathbf{Y}_i$ to denote the vector of y-values in case (i), we have

$$\mathbf{Y}_1 = \begin{bmatrix} -4 \\ -2 \\ 0 \\ 2 \\ 4 \end{bmatrix}, \quad \mathbf{Y}_2 = \begin{bmatrix} -5 \\ 3 \\ 1 \\ -3 \\ 4 \end{bmatrix}, \quad \mathbf{Y}_3 = \begin{bmatrix} 6 \\ 2 \\ -1 \\ -2 \\ -5 \end{bmatrix}, \quad \mathbf{Y}_4 = \begin{bmatrix} 4 \\ -2 \\ -1 \\ 3 \\ -4 \end{bmatrix},$$

leading to the respective dot products $\mathbf{Y}_1 \cdot \mathbf{X} = 20$, $\mathbf{Y}_2 \cdot \mathbf{X} = 12$, $\mathbf{Y}_3 \cdot \mathbf{X} = -26$, $\mathbf{Y}_4 \cdot \mathbf{X} = -11$, so the best-fit lines are given by the respective equations

$$y = 2x, \quad y = (6/5)x, \quad y = -(13/5)x, \quad y = -(11/10)x.$$



$y = 2x$
$r = 1$

$y = (6/5)x$
$r \approx .490$

$y = -(13/5)x$
$r \approx -.983$

$y = -(11/10)x$
$r \approx -.513$

FIGURE 7.3.4. The best-fit lines for Example 7.3.4.

The correlation coefficient r (found in Example 2.4.8) is close to 1 or -1 in the first and third cases, and not so close to those options in the other two cases. This matches Figure 7.3.4: in the first and third cases the best fit line is a good fit with the data, but in the second and fourth cases it is a poor fit.

This should not come as a surprise: we already mentioned in Theorem 2.4.5 that the closeness of r to ± 1 (or equivalently the closeness of r^2 to 1) tells us how well the best fit line matches the data. This can be made more mathematically precise as follows. Let the best-fit line be $y = mx + b$, and let r be the correlation coefficient for the recentered data $(x_i - \overline{x}, y_i - \overline{y})$ (whose coordinates average to 0 by Remark 2.4.4) with associated n-vectors $\widehat{\mathbf{X}}$ and $\widehat{\mathbf{Y}}$. Then the role of nearness of r^2 to 1 (or equivalently of nearness of $1 - r^2$ to 0) as a measure of quality of fit is expressed by the following identity (*which you shouldn't memorize*):

$$\|\mathbf{Y} - (m\mathbf{X} + b\mathbf{1})\|^2 = \|\widehat{\mathbf{Y}}\|^2\,(1 - r^2), \tag{7.3.5}$$

where $\widehat{\mathbf{Y}}$ is the "recentered" version of $\mathbf{Y}$ (subtracting $\overline{y}$ from all y_i's).

To explain the meaning of (7.3.5), expand out the left side (and use that $t^2 = |t|^2$ for any t) to get

$$|y_1 - (mx_1 + b)|^2 + |y_2 - (mx_2 + b)|^2 + \cdots + |y_n - (mx_n + b)|^2.$$

The number $|y_i - (mx_i + b)|$ is the *vertical distance* between the data point (x_i, y_i) and the best fit line $y = mx + b$. When $r^2 \approx 1$, (7.3.5) therefore says that these vertical distances are "collectively small": the sum of their squares is tiny since $1 - r^2$ on the right side of (7.3.5) is small, so the data points are *all close* to the best fit line. When $r^2 \approx 0$ then (at least informally) the opposite happens since the right side is approximately $\|\widehat{\mathbf{Y}}\|^2$, which is typically quite far from 0 (even though the average of the entries in $\widehat{\mathbf{Y}}$ is 0 by design). For those who are interested, a proof of (7.3.5) is given in Section 7.5 below. ■

7.4. Distance between two lines.

We next discuss a problem that can be answered via multivariable optimization techniques to be discussed later but has an elegant solution using insights from linear algebra via Theorem 7.1.1: how do we find the closest distance between two lines in $\mathbf{R}^3$ (that do not have to pass through the origin) or in $\mathbf{R}^n$? The version in $\mathbf{R}^3$ is important for collision detection algorithms which ensure distances stay above a given threshold (e.g., so objects with known diameters moving along straight lines in various directions won't collide).

Example 7.4.1. Let ℓ_1 be the line in $\mathbf{R}^3$ that is parallel to $\mathbf{v} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}$ and goes through $\mathbf{a} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, and let ℓ_2 be the line in $\mathbf{R}^3$ that is parallel to $\mathbf{w} = \begin{bmatrix} 4 \\ 3 \\ 2 \end{bmatrix}$ and goes through $\mathbf{b} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$. These are shown in Figure 7.4.1. What is the closest distance between ℓ_1 and ℓ_2?

The idea is to rewrite this problem involving lines in $\mathbf{R}^3$ as a question of finding a closest point to a *plane*, as we now explain. The parametric form of lines in space (see Example 3.3.1) gives that ℓ_1 consists of the points of the form $\mathbf{a} + t\mathbf{v}$ for an arbitrary scalar t, and ℓ_2 consists of the points of the form $\mathbf{b} + t'\mathbf{w}$ for an arbitrary scalar t'. The distance between any two such points is given by $\|\,(\mathbf{a} + t\mathbf{v}) - (\mathbf{b} + t'\mathbf{w})\,\|$, and we seek t, t' that minimize it. This is a geometry problem that one may initially think about in terms of the picture in Figure 7.4.1.

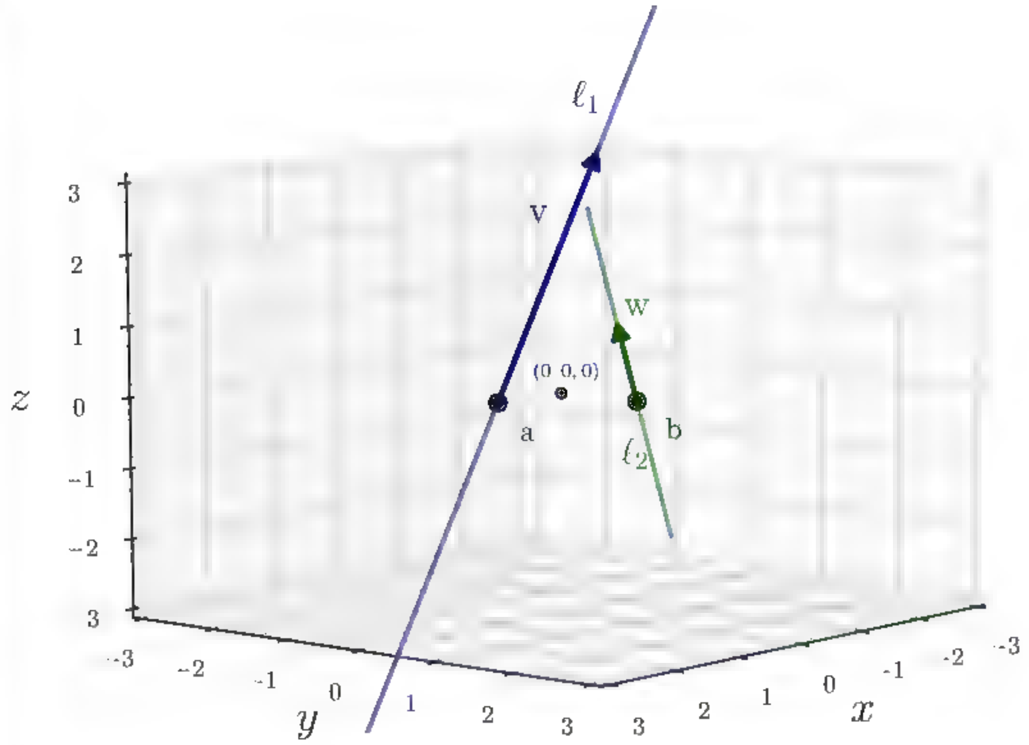


FIGURE 7.4.1. Lines ℓ_1 through $\mathbf{a}$ in the direction $\mathbf{v}$ and ℓ_2 through $\mathbf{b}$ in the direction $\mathbf{w}$

Now comes the key step: since $(\mathbf{a}+t\mathbf{v})-(\mathbf{b}+t'\mathbf{w})=(\mathbf{a}-\mathbf{b})-(t'\mathbf{w}-t\mathbf{v})$, the length we are trying to minimize (as t and t' vary) can also be interpreted as the distance between the vector $\mathbf{a}-\mathbf{b}$ and the vector $t'\mathbf{w}-t\mathbf{v}$. The points $\mathbf{a}-\mathbf{b}$ and $t'\mathbf{w}-t\mathbf{v}$ are shown in Figure 7.4.2.

As we vary over all values of the scalars t and t', by definition of "span" the vectors $t'\mathbf{w}-t\mathbf{v}$ sweep out the entire plane $\text{span}(\mathbf{v},\mathbf{w})$ (this is a plane since $\mathbf{v}$ and $\mathbf{w}$ are nonzero and not scalar multiples of each other). Hence, as illustrated in Figure 7.4.3, our problem now may be restated in an alternative manner: what is the shortest distance between the point $\mathbf{a}-\mathbf{b}\in\mathbf{R}^3$ and the plane $V=\text{span}(\mathbf{v},\mathbf{w})$?

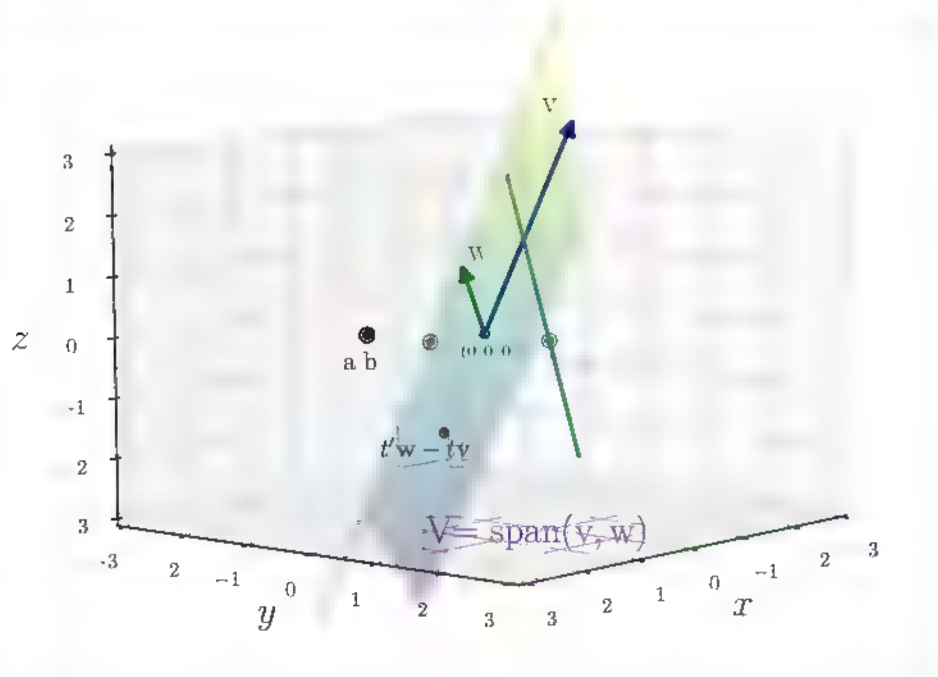


FIGURE 7.4.2. The point $\mathbf{a}-\mathbf{b}$ and a point $t'\mathbf{w}-t\mathbf{v}$ in the plane $V=\text{span}(\mathbf{v},\mathbf{w})$

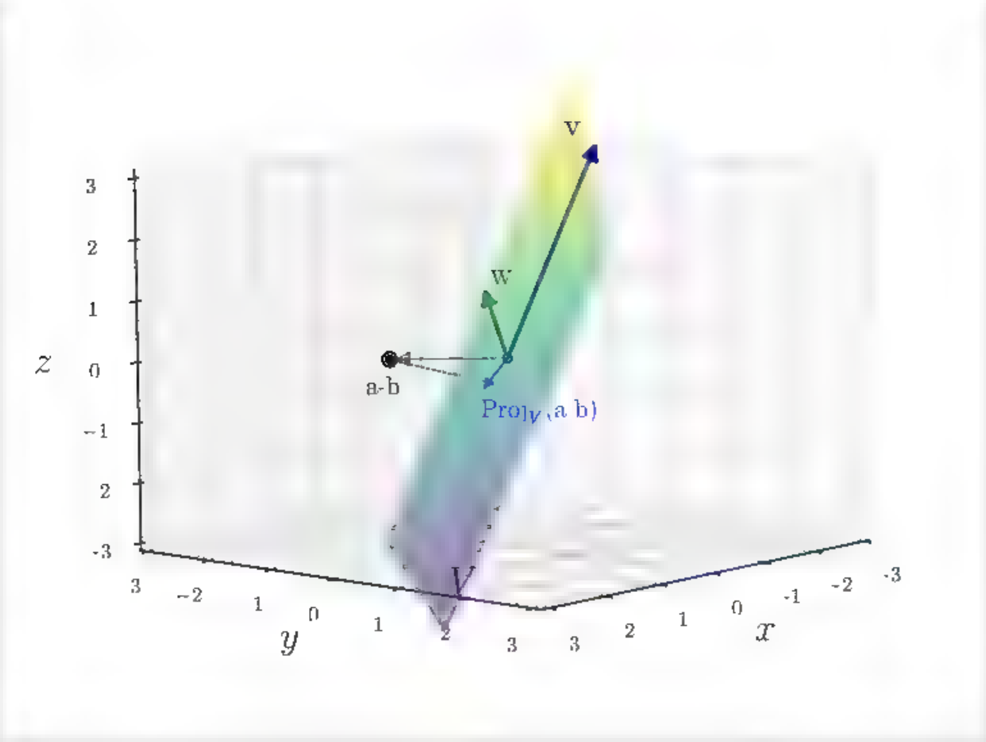


FIGURE 7.4.3. Reformulating distance between lines as distance from point to plane!

We have used vector algebra to transform the original geometric problem involving the closest distance between two lines into the rather different-looking geometric problem of the distance from a point to a plane! It is rather amazing how a bit of algebra can transform one type of geometry problem into a very different-looking geometry problem. This illustrates why the synthesis of algebra and geometry in linear algebra is so powerful.

The advantage of this reformulation is that we know how to solve it: the point of V closest to $\mathbf{a}-\mathbf{b}$ is the projection of $\mathbf{a}-\mathbf{b}$ into V (at the end of the dotted segment in Figure 7.4.3), so we shall compute this projection and then find its distance to $\mathbf{a}-\mathbf{b}$ to get our desired answer.

Let's formulate the general solution method, and then carry it out in the specific situation given above. Although the multi-step method may look like a mouthful, in the worked illustration of it below for the preceding situation you'll see that it is very clean and systematic, and so doesn't require any memorization of new formulas. (As written, the method works for a pair of parametric lines in $\mathbf{R}^n$ for any n.)

Step 1 Use Theorem 7.1.1 to compute an orthogonal basis $\{\mathbf{w}, \mathbf{v}'\}$ of $V = \mathrm{span}(\mathbf{v}, \mathbf{w})$ from the given spanning set $\{\mathbf{v}, \mathbf{w}\}$ (we can also replace $\mathbf{v}'$ with any convenient nonzero scalar multiple, or use the orthogonal basis $\{\mathbf{v}, \mathbf{w}'\}$ instead, and adjust what follows accordingly).

Step 2 Compute the projection

$$\mathbf{Proj}_V(\mathbf{a}-\mathbf{b}) = \frac{(\mathbf{a}-\mathbf{b})\cdot\mathbf{w}}{\mathbf{w}\cdot\mathbf{w}}\mathbf{w} + \frac{(\mathbf{a}-\mathbf{b})\cdot\mathbf{v}'}{\mathbf{v}'\cdot\mathbf{v}'}\mathbf{v}' = c\mathbf{w} + c'\mathbf{v}'$$

with the coefficients c and c' given by the indicated ratios of dot products. *Don't compute* $c\mathbf{w} + c'\mathbf{v}'$ *as an explicit vector yet*; leave it in the form of a "symbolic" linear combination of $\mathbf{w}$ and $\mathbf{v}'$ with explicit coefficients. Keeping $\mathbf{v}'$ and $\mathbf{w}$ in symbolic form (rather than as explicit numerical vectors) will make the subsequent algebra cleaner.

Step 3 Rewrite $c\mathbf{w} + c'\mathbf{v}'$ in terms of the original $\mathbf{v}$ and $\mathbf{w}$ by substituting in place of $\mathbf{v}'$ its definition $\mathbf{v} - \mathbf{Proj}_{\mathbf{w}}(\mathbf{v}) = \mathbf{v} - \dfrac{\mathbf{v}\cdot\mathbf{w}}{\mathbf{w}\cdot\mathbf{w}}\mathbf{w}$ in terms of $\mathbf{v}$ and $\mathbf{w}$. Combine all $\mathbf{v}$-terms and all $\mathbf{w}$-terms to arrive at an expression $r\mathbf{v} + s\mathbf{w}$ for $c\mathbf{w} + c'\mathbf{v}'$ with some explicit scalars r and s. That is, we have computed $\mathbf{Proj}_V(\mathbf{a}-\mathbf{b}) = r\mathbf{v} + s\mathbf{w}$ for some explicit r and s.

Step 4 We already know that the point $\mathbf{Proj}_V(\mathbf{a}-\mathbf{b}) \in V$ nearest to $\mathbf{a}-\mathbf{b}$ has the form $t'\mathbf{w}-t\mathbf{v}$ for the unknown scalars t and t' making $\mathbf{a}+t\mathbf{v}$ and $\mathbf{b}+t'\mathbf{w}$ be the points of minimal distance on the two lines. By Step 3, we conclude that $r\mathbf{v}+s\mathbf{w} = t'\mathbf{w}-t\mathbf{v} = -t\mathbf{v}+t'\mathbf{w}$, so (since $\mathbf{v}$ and $\mathbf{w}$ are not scalar multiples of each other) the coefficients must match: $-t = r$ (so $t = -r$) and $t' = s$. We have solved for t and t'! In other words, $\mathbf{a} - r\mathbf{v}$ and $\mathbf{b} + s\mathbf{w}$ are the points at minimal distance (and these can be explicitly computed, as can the distance between them, if we wish).

Returning to our specific situation of interest, for Step 1 the basis $\{\mathbf{v}, \mathbf{w}\}$ of V yields the orthogonal basis $\{\mathbf{w}, \mathbf{v}'\}$ with

$$\mathbf{w} = \begin{bmatrix} 4 \\ 3 \\ 2 \end{bmatrix}, \quad \mathbf{v}' = \mathbf{v} - \mathbf{Proj}_{\mathbf{w}}\,\mathbf{v} = \mathbf{v} - \frac{\mathbf{v}\cdot\mathbf{w}}{\mathbf{w}\cdot\mathbf{w}}\mathbf{w} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix} - \frac{25}{29}\begin{bmatrix} 4 \\ 3 \\ 2 \end{bmatrix} = \frac{1}{29}\begin{bmatrix} -42 \\ 12 \\ 66 \end{bmatrix}.$$

The scaling factor $1/29$ is annoying (though it cancels out in any projection calculations as done below), so we'll replace $\mathbf{v}'$ with the scalar multiple

$$\mathbf{v}'' = 29\mathbf{v}' = \begin{bmatrix} -42 \\ 12 \\ 66 \end{bmatrix};$$

that is, $\{\mathbf{w}, \mathbf{v}''\}$ is also an orthogonal basis of V and we will use it to compute $\mathbf{Proj}_V$.

Moving on to Step 2, the projection of $\mathbf{a} - \mathbf{b} = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$ into V is given by

$$\mathbf{Proj}_{\mathbf{w}}\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} + \mathbf{Proj}_{\mathbf{v}''}\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} = \frac{1}{29}\mathbf{w} + \frac{-1}{116}\mathbf{v}''.$$

This completes Step 2. Next carrying out Step 3, we express this in terms of $\mathbf{v}$ and $\mathbf{w}$ by going back to how $\mathbf{v}''$ was built in terms of $\mathbf{v}'$ and especially how $\mathbf{v}'$ was built in terms of $\mathbf{v}$ and $\mathbf{w}$:

$$\begin{aligned} \frac{1}{29}\mathbf{w} - \frac{1}{116}\mathbf{v}'' = \frac{1}{29}\mathbf{w} - \frac{1}{116}(29\mathbf{v}') &= \frac{1}{29}\mathbf{w} - \frac{1}{4}\mathbf{v}' \\ &= \frac{1}{29}\mathbf{w} - \frac{1}{4}(\mathbf{v} - \mathbf{Proj}_{\mathbf{w}}(\mathbf{v})) \\ &= \frac{1}{29}\mathbf{w} - \frac{1}{4}(\mathbf{v} - \frac{\mathbf{v}\cdot\mathbf{w}}{\mathbf{w}\cdot\mathbf{w}}\mathbf{w}) \\ &= \frac{1}{29}\mathbf{w} - \frac{1}{4}(\mathbf{v} - \frac{25}{29}\mathbf{w}) \\ &= -\frac{1}{4}\mathbf{v} + \frac{1}{4}\mathbf{w}. \end{aligned}$$

This completes Step 3.

Finally, we remember that this projection we have computed also has the form $t'\mathbf{w} - t\mathbf{v}$ for the unknown scalars t and t' making $\mathbf{a} + t\mathbf{v}$ and $\mathbf{b} + t'\mathbf{w}$ have minimal distance, so we solve for these scalars by equating it with the expression obtained at the end of Step 3 and comparing coefficients. That is, we have $t'\mathbf{w}-t\mathbf{v} = -(1/4)\mathbf{v}+(1/4)\mathbf{w}$, so $t' = 1/4$ and $t = 1/4$. Hence, the points achieving the minimal distance are

$$\mathbf{a}+\frac{1}{4}\mathbf{v}=\begin{bmatrix}1\\0\\0\end{bmatrix}+\frac{1}{4}\begin{bmatrix}2\\3\\4\end{bmatrix}=\begin{bmatrix}3/2\\3/4\\1\end{bmatrix},\quad \mathbf{b}+\frac{1}{4}\mathbf{w}=\begin{bmatrix}0\\1\\0\end{bmatrix}+\frac{1}{4}\begin{bmatrix}4\\3\\2\end{bmatrix}=\begin{bmatrix}1\\7/4\\1/2\end{bmatrix}.$$

The distance between these points is the length of the difference between these points, which is $\begin{bmatrix}1/2\\-1\\1/2\end{bmatrix}$ (or its negative, depending on the order of subtraction). This length is $\sqrt{1/4+1+1/4}=\sqrt{3/2}$. ■

7.5. Orthogonal basis formula and relation of correlation coefficient to best fit lines. In this section we prove some results discussed in this chapter, beginning with a proof of Theorem 7.1.1.

PROOF. Write $\mathbf{x}'=\mathbf{x}-\mathbf{Proj}_{\mathbf{y}}\,\mathbf{x}$. We saw $\mathbf{x}'\cdot\mathbf{y}=0$ in the proof of Proposition 6.1.1, but let's see it again:

$$\mathbf{x}'\cdot\mathbf{y}=\left(\mathbf{x}-\frac{\mathbf{x}\cdot\mathbf{y}}{\mathbf{y}\cdot\mathbf{y}}\mathbf{y}\right)\cdot\mathbf{y}=\mathbf{x}\cdot\mathbf{y}-\frac{\mathbf{x}\cdot\mathbf{y}}{\mathbf{y}\cdot\mathbf{y}}\mathbf{y}\cdot\mathbf{y}=\mathbf{x}\cdot\mathbf{y}-\mathbf{x}\cdot\mathbf{y}=0. \tag{7.5.1}$$

Next, $\mathbf{y}$ is not zero (we have assumed this). Also, $\mathbf{x}'$ is not zero: if it were zero then $\mathbf{x}=\mathbf{Proj}_{\mathbf{y}}(\mathbf{x})$, yet such a projection is always scalar multiple of $\mathbf{y}$ and we have assumed $\mathbf{x}$ is not a scalar multiple of $\mathbf{y}$. Therefore $\mathbf{x}',\mathbf{y}$ is a pair of nonzero orthogonal vectors belonging to $\mathrm{span}(\mathbf{x},\mathbf{y})$ by design (note that $\mathbf{y}=0\,\mathbf{x}+1\,\mathbf{y}$), and they exhaust that span since we can also write each of $\mathbf{x}$ and $\mathbf{y}$ as linear combinations of $\mathbf{x}'$ and $\mathbf{y}$: $\mathbf{x}=\mathbf{x}'+\mathbf{Proj}_{\mathbf{y}}(\mathbf{x})=\mathbf{x}'+((\mathbf{x}\cdot\mathbf{y})/(\mathbf{y}\cdot\mathbf{y}))\mathbf{y}$ and $\mathbf{y}=0\,\mathbf{x}+1\,\mathbf{y}$. Since any collection of pairwise orthogonal nonzero vectors is a basis for its span (Theorem 5.2.2), we conclude that $\{\mathbf{x}',\mathbf{y}\}$ is an orthogonal basis of $\mathrm{span}(\mathbf{x}',\mathbf{y})=\mathrm{span}(\mathbf{x},\mathbf{y})$. ☐

Now suppose we are given n data points (x_i,y_i), assembled into n-vectors

$$\mathbf{X}=\begin{bmatrix}x_1\\x_2\\\vdots\\x_n\end{bmatrix}\text{ and }\mathbf{Y}=\begin{bmatrix}y_1\\y_2\\\vdots\\y_n\end{bmatrix}.$$

In Theorem 2.4.5 we described the relationship between the correlation coefficient r for the recentered data (corresponding to the n-vectors $\widehat{\mathbf{X}}$ and $\widehat{\mathbf{Y}}$) and the line of best fit. Let's restate that in terms of r^2, which we expressed in (2.4.2) as the formula

$$r^2=\frac{(\widehat{\mathbf{X}}\cdot\widehat{\mathbf{Y}})^2}{\|\widehat{\mathbf{X}}\|^2\|\widehat{\mathbf{Y}}\|^2}=\frac{(\widehat{\mathbf{X}}\cdot\widehat{\mathbf{Y}})^2}{(\widehat{\mathbf{X}}\cdot\widehat{\mathbf{X}})(\widehat{\mathbf{Y}}\cdot\widehat{\mathbf{Y}})}.$$

We stated that r^2 is near 0 when the line of best fit is a bad fit, and near 1 when it is a good fit (note that this could happen either when r is near 1, or when r is near -1). We made the role of r^2 as a measure of quality of fit precise with (7.3.5). Here is the derivation of (7.3.5); it is a good illustration of the utility of vector algebra.

PROOF. Using (7.3.3) we have

$$\mathbf{Y}-(m\mathbf{X}+b\mathbf{1})=\mathbf{Y}-\left(\left(\frac{\mathbf{Y}\cdot\widehat{\mathbf{X}}}{\widehat{\mathbf{X}}\cdot\widehat{\mathbf{X}}}\right)\widehat{\mathbf{X}}+\left(\frac{\mathbf{Y}\cdot\mathbf{1}}{\mathbf{1}\cdot\mathbf{1}}\right)\mathbf{1}\right)=\underbrace{\left(\mathbf{Y}-\left(\frac{\mathbf{Y}\cdot\mathbf{1}}{\mathbf{1}\cdot\mathbf{1}}\right)\mathbf{1}\right)}_{=\widehat{\mathbf{Y}}}-\left(\frac{\mathbf{Y}\cdot\widehat{\mathbf{X}}}{\widehat{\mathbf{X}}\cdot\widehat{\mathbf{X}}}\right)\widehat{\mathbf{X}}, \tag{7.5.2}$$

where $\mathbf{Y} - \left(\frac{\mathbf{Y}\cdot\mathbf{1}}{\mathbf{1}\cdot\mathbf{1}}\right)\mathbf{1} = \mathbf{Y} - \overline{y}\mathbf{1}$ is indeed equal to $\widehat{\mathbf{Y}}$.

Note that $\widehat{\mathbf{Y}}\cdot\widehat{\mathbf{X}} = \mathbf{Y}\cdot\widehat{\mathbf{X}}$ because the difference $\widehat{\mathbf{Y}} - \mathbf{Y} = -\overline{y}\mathbf{1}$ is orthogonal to $\widehat{\mathbf{X}}$. (Please make sure you understand why this is true.) Putting this into the numerator of the final coefficient on the right side of (7.5.2) yields

$$\mathbf{Y} - (m\mathbf{X} + b\mathbf{1}) = \widehat{\mathbf{Y}} - \left(\frac{\widehat{\mathbf{Y}}\cdot\widehat{\mathbf{X}}}{\widehat{\mathbf{X}}\cdot\widehat{\mathbf{X}}}\right)\widehat{\mathbf{X}} = \widehat{\mathbf{Y}} - \mathbf{Proj}_{\widehat{\mathbf{X}}}\,\widehat{\mathbf{Y}};$$

the final equality is due to the formula (6.1.2) for projection to the span of a nonzero vector.

The vectors $\widehat{\mathbf{Y}} - \mathbf{Proj}_{\widehat{\mathbf{X}}}\,\widehat{\mathbf{Y}}$ and $\mathbf{Proj}_{\widehat{\mathbf{X}}}\,\widehat{\mathbf{Y}}$ are perpendicular to each other (you can either check this with vector algebra, which is exactly what is done in (7.5.1), or draw a picture). Therefore, by the Pythagorean Theorem in $\mathbf{R}^n$ (see Theorem 2.3.1) we have

$$\|\widehat{\mathbf{Y}}\|^2 = \|(\widehat{\mathbf{Y}} - \mathbf{Proj}_{\widehat{\mathbf{X}}}\,\widehat{\mathbf{Y}}) + \mathbf{Proj}_{\widehat{\mathbf{X}}}\,\widehat{\mathbf{Y}}\|^2 = \|\widehat{\mathbf{Y}} - \mathbf{Proj}_{\widehat{\mathbf{X}}}\,\widehat{\mathbf{Y}}\|^2 + \|\mathbf{Proj}_{\widehat{\mathbf{X}}}\,\widehat{\mathbf{Y}}\|^2,$$

so $\|\widehat{\mathbf{Y}} - \mathbf{Proj}_{\widehat{\mathbf{X}}}\,\widehat{\mathbf{Y}}\|^2 = \|\widehat{\mathbf{Y}}\|^2 - \|\mathbf{Proj}_{\widehat{\mathbf{X}}}\,\widehat{\mathbf{Y}}\|^2$. But the vector difference on the left side is exactly $\mathbf{Y} - (m\mathbf{X} + b\mathbf{1})$ by (7.5.2), so

$$\|\mathbf{Y} - (m\mathbf{X} + b\mathbf{1})\|^2 = \|\widehat{\mathbf{Y}}\|^2 - \|\mathbf{Proj}_{\widehat{\mathbf{X}}}\,\widehat{\mathbf{Y}}\|^2. \tag{7.5.3}$$

Finally, using the definition of $\mathbf{Proj}_{\widehat{\mathbf{X}}}\,\widehat{\mathbf{Y}}$, we have

$$\|\mathbf{Proj}_{\widehat{\mathbf{X}}}\,\widehat{\mathbf{Y}}\|^2 = \left(\frac{\widehat{\mathbf{Y}}\cdot\widehat{\mathbf{X}}}{\widehat{\mathbf{X}}\cdot\widehat{\mathbf{X}}}\right)\widehat{\mathbf{X}}\cdot\left(\frac{\widehat{\mathbf{Y}}\cdot\widehat{\mathbf{X}}}{\widehat{\mathbf{X}}\cdot\widehat{\mathbf{X}}}\right)\widehat{\mathbf{X}} = \left(\frac{\widehat{\mathbf{Y}}\cdot\widehat{\mathbf{X}}}{\widehat{\mathbf{X}}\cdot\widehat{\mathbf{X}}}\right)^2\widehat{\mathbf{X}}\cdot\widehat{\mathbf{X}} = \frac{(\widehat{\mathbf{Y}}\cdot\widehat{\mathbf{X}})^2}{\widehat{\mathbf{X}}\cdot\widehat{\mathbf{X}}} = r^2(\widehat{\mathbf{Y}}\cdot\widehat{\mathbf{Y}}),$$

so plugging into (7.5.3) yields $\|\mathbf{Y} - (m\mathbf{X} + b\mathbf{1})\|^2 = \|\widehat{\mathbf{Y}}\|^2(1 - r^2)$, which is exactly the desired identity (7.3.5). □

Chapter 7 highlights (links to highlights in [illegible] and next chapters)

Notation	**Meaning**	**Location in text**
the n-vector $\mathbf{1}$	the n-vector whose entries are all equal to 1 (analogous to $\mathbf{0} \in \mathbf{R}^n$)	discussion preceding (7.3.2)

Concept	**Meaning**	**Location in text**
best fit line to data in $\mathbf{R}^2$ (in "least squares" sense)	line in $\mathbf{R}^2$ whose sum of squares of (vertical!) errors from given data points is minimized	Remark 7.3.1

Result	**Meaning**	**Location in text**
construction of orthogonal basis of a "plane" in $\mathbf{R}^n$	given a basis for a 2-dimensional linear subspace V in $\mathbf{R}^n$, an explicit recipe to produce an orthogonal basis for V (useful for computing $\mathbf{Proj}_V$)	Theorem 7.1.1
interpret linear regression for n data points in $\mathbf{R}^2$ (not on a common vertical line) as distance minimization in $\mathbf{R}^n$	projection of $\mathbf{Y}$ into $\mathrm{span}(\mathbf{X}, \mathbf{1})$ is $m\mathbf{X} + b\mathbf{1}$, with "best fit" line being $y = mx + b$	(7.3.1)

Skill	**Location in text**
for 2-dimensional linear subspace V in $\mathbf{R}^n$, compute orthogonal basis of V from a given basis	Example 7.1.2 (for the method; examples on homework and exams will always involve integers or fractions with small denominators, no need for a calculator)
compute line of best fit to n data points in $\mathbf{R}^2$ (not on a common vertical line) via projection to a specific plane in $\mathbf{R}^n$ determined by the data	Example 7.3.2, Example 7.3.3, Example 7.3.4

Part II

Multivariable functions and optimization

“Nothing takes place in the world whose meaning is not that of some maximum or minimum.”

L. Euler

“Fortunately, using linear methods well known to me from [advanced math], I was able to build a simplified approximate model for much of the 10-dimensional expectation surface. My methods reduced the problem from 400 million years [of computation time] to [. . .] about 2 hours of actual computer time.”

E. Thorp [Th, p. 34], author of *Beat the Dealer*

Overview of Part II

In this second Part, comprising Chapters 8–12, we get to what you may have considered to be the main theme of the course (before learning about the role of linear algebra): functions of several variables and how to do differential calculus with them. The experience from Part I with lines and planes in both $\mathbf{R}^3$ and $\mathbf{R}^n$ for large n is useful for working with functions $f(x_1, \ldots, x_n)$ of n variables. We shall extend the idea of derivatives to such functions and discuss its applications.

Chapter 8 introduces the idea of multivariable functions, and how to visualize them. There are a few novel aspects. For example, we usually consider functions of one variable as being defined on the entire real line, or on some interval. But the region on which a function of two variables can be defined in $\mathbf{R}^2$ can have rather complicated geometry. We do not delve into this, but that is a genuinely new phenomenon. Visualizing functions of two or more variables is another new challenge. You might be tempted to look at their graphs, as for functions of one variable, but this is not the only useful way: we will see how level sets of such functions provide another good visualization. As with n-vectors and general linear subspaces of $\mathbf{R}^n$ and all the other things you learned in Part I, for functions of 4 or more variables we can no longer literally visualize them; instead we use the geometric language of “higher-dimensional” linear algebra to extend our intuition.

There are several notions which might be called the “derivative(s)” of a multivariable function. In Chapter 9 we introduce the most immediate notion: *partial derivatives*. The idea is that we consider a function $f(x_1, \ldots, x_n)$ of n variables as a collection of functions of one variable x_i by fixing all but the ith variable and then using single-variable calculus with the remaining variable x_i. This underlies one of the main applications of multivariable calculus, discussed in Chapter 10: finding candidates for where a function f of n variables attains its maximum and minimum values via a “first-derivative test” similar to the single-variable case.

In Chapter 11 we introduce a slightly more general notion of derivative, called the *gradient* of a function $f : \mathbf{R}^n \rightarrow \mathbf{R}$. It is a vector $(\nabla f)(\mathbf{a})$ at every point $\mathbf{a} \in \mathbf{R}^n$, telling us how f is varying near $\mathbf{a}$. The gradient descent method described in Chapter 11 is important in real-world optimization problems. The Lagrange multiplier method in Chapter 12, for optimizing functions $\mathbf{R}^n \rightarrow \mathbf{R}$ whose variables are constrained by some further relationship, also uses the concept of gradient.

Optimization for functions of n variables arises in many places in natural sciences, engineering, economics, and elsewhere. Because of this great variety of applications, the subject of optimization has taken on a life of its own; people from different disciplines think about it in rather different ways. The underlying ideas all stem from the ones presented in this Part, but the specific contexts in which these problems arise lead to specialized techniques tailored to fit special circumstances. Linear programming, convex optimization, the principle of least action in physics, and equilibria that arise in economic transactions are all examples of multivariable optimization in practice.

Although you will encounter optimization in different guises in many other courses, the ideas that you learn here constitute the core principles of the subject and are well worth mastering.

8. Multivariable functions, level sets, and contour plots

Functions of more than one variable are called *multivariable functions*. Calculus that you studied before this course involves functions of one variable, but many real-world quantitative phenomena (in economics, natural sciences, computer programming, data science, and so much more) depend on many unknowns. Thus, multivariable functions are a fundamental concept in applications of mathematics and we need to adapt our visualization skills with single-variable functions (such as via graphs) to the multivariable case. This chapter focuses on defining and illustrating a lot of useful new terminology.

By the end of this chapter, you should be able to:

- distinguish between scalar and vector-valued multivariable functions;
- determine the component functions of a vector-valued function;
- recognize the level sets of a multivariable function;
- interpret some basic features of a 2-variable function (directions of increase and decrease) from a plot of its level sets (i.e., a contour plot).

8.1. Basic terminology. For useful notation (e.g., $f : A \to B$) and terminology involving functions, see Table 0.0.1 and Appendix A. In particular a *function from* $\mathbf{R}^n$ *to* $\mathbf{R}^m$, typically denoted as

$$\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m,$$

takes vectors in $\mathbf{R}^n$ as *input* and gives vectors in $\mathbf{R}^m$ as *output*. Keep in mind that *a function assigns to each input a* **single** *output, but it is fine if two inputs yield the same output* (e.g., $\mathbf{f}(\mathbf{x}) = \|\mathbf{x}\| = \mathbf{f}(-\mathbf{x})$).

Definition 8.1.1. A *scalar-valued* function is a function $\mathbf{R}^n \to \mathbf{R}$ (that is to say, with $m = 1$). In other words, a scalar-valued function gives real number outputs.

We have already seen a scalar-valued multivariable function in Section 7.3: the problem of finding a best-fit line involved minimizing a scalar-valued function $\mathrm{E} : \mathbf{R}^2 \to \mathbf{R}$ of the vector $(m, b) \in \mathbf{R}^2$ (or more concretely, E is an $\mathbf{R}$-valued function of two variables m and b). Here are some more examples of scalar-valued functions.

Example 8.1.2. The gas mileage you get from your car depends on the temperature T, and also on the tire pressure P. Thus your mileage M can be regarded as a scalar-valued function $M(T, P)$ of T and P. ■

Example 8.1.3. The solubility σ of air in water depends on the temperature T and the salinity S of the water. Therefore solubility can be regarded as a scalar-valued function $\sigma(T, S)$ of T and S. ■

In both of the preceding examples, the function in question – mileage or solubility – also depends on many other parameters; for example, mileage also depends on the number of passengers in the car. Realistically we should regard mileage as a function $M(T, P)$ of just the temperature T and pressure P only in a situation where these other parameters are held constant.

Example 8.1.4. Addition and multiplication are scalar-valued functions $\mathbf{R}^2 \to \mathbf{R}$:

$$A(x_1, x_2) = x_1 + x_2 \qquad M(x_1, x_2) = x_1 x_2.$$

■

Example 8.1.5. If (x_1, x_2) is a point in $\mathbf{R}^2$, the distance between $(0, 0)$ and (x_1, x_2) is given by the formula $\sqrt{x_1^2 + x_2^2}$. You can also think of this as the *length* of the vector with coordinates x_1 and x_2; that is,

$$\|\ \| : \mathbf{R}^2 \to \mathbf{R} \quad \text{defines a function, where} \quad \left\| \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right\| = \sqrt{x_1^2 + x_2^2}.$$

Therefore "length of a vector in $\mathbf{R}^2$" is a scalar-valued function. ■

Example 8.1.6. Define scalar-valued functions $f : \mathbf{R}^3 \to \mathbf{R}$ and $g : \mathbf{R}^3 \to \mathbf{R}$ by the rules

$$f(\mathbf{x}) = \mathbf{x} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad g(\mathbf{x}) = \mathbf{x} \cdot \begin{bmatrix} -1 \\ 3 \\ 0 \end{bmatrix}.$$

(Such rules arise when computing the projection of vectors onto a fixed line.) Written out explicitly, $f\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = x + y + z$ and $g\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = -x + 3y$. ■

Definition 8.1.7. A *vector-valued* function is a function $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ with general $m \geq 1$. In other words, a vector-valued function gives output considered as vectors in some $\mathbf{R}^m$.

A vector-valued function $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ can be expressed in terms of m scalar-valued *component functions* or *coordinate functions* $f_1, \dots, f_m : \mathbf{R}^n \to \mathbf{R}$, defined by the expressions

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix} = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$$

(depending on whether we consider the output to be a "vector" or a "point"), with each f_j a scalar-valued function.

We can write the output of $\mathbf{f}$ on the input $\mathbf{x} \in \mathbf{R}^n$ in (at least) three ways:

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}\left(\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \mathbf{f}(x_1, \dots, x_n),$$

depending on whether we want to keep things compact, emphasize that the input to $\mathbf{f}$ is considered as a vector in $\mathbf{R}^n$, or emphasize that the output of $\mathbf{f}$ depends on n real-number inputs (the coordinates of the point or vector $\mathbf{x}$).

Example 8.1.8. Consider a small object flying through the air. At any given time t, its position in space is a point $\mathbf{x}(t) = (x(t), y(t), z(t)) \in \mathbf{R}^3$ and its velocity (a vector pointing in the direction of motion with magnitude equal to the speed) is some

$$\mathbf{v}(t) = \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} \in \mathbf{R}^3,$$

so both position and velocity are $\mathbf{R}^3$-valued functions of time $t \in \mathbf{R}$. In other words, we have vector-valued functions

$$\mathbf{x} : \mathbf{R} \to \mathbf{R}^3, \quad \mathbf{v} : \mathbf{R} \to \mathbf{R}^3.$$

In physics one learns that the component functions of $\mathbf{x}$ and $\mathbf{v}$ are linked through differentiation:

$$v_1(t) = x'(t), \quad v_2(t) = y'(t), \quad v_3(t) = z'(t).$$

If we differentiate the component functions once more, we get yet another vector-valued function

$$\mathbf{a} : \mathbf{R} \to \mathbf{R}^3$$

given by $\mathbf{a}(t) = (v_1'(t), v_2'(t), v_3'(t)) = (x''(t), y''(t), z''(t))$ called the *acceleration* (which is the focus of one of Newton's force laws). ■

Example 8.1.9. Suppose that we want to describe the wind throughout the San Francisco Bay Area at a certain time. We can describe the wind velocity, at any given point, by a 2-vector measuring the wind velocities (say in miles per hour) in the north and east direction. For example, if the wind velocity at some point is $\mathbf{v} = \begin{bmatrix} 10 \\ 10 \end{bmatrix} = 10 \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, that means the wind speed there is $\|\mathbf{v}\| = 10\sqrt{2} \approx 14.14$ miles per hour, with the wind there blowing in exactly a northeast direction.

The wind speed, as well as the wind direction, varies substantially across the San Francisco region (because of the influence of mountains, bodies of water, etc.) To describe it completely, then, we must give the velocity $\mathbf{v}$ at each point. Thus, if we introduce a coordinate system (x, y) – where x is the number of miles east of Stanford, and y the number of miles north – we can describe the wind by giving a function

$$\mathbf{v}(x, y) = \text{ wind velocity at position } (x, y).$$

The function $\mathbf{v}$ is a typical example of a multivariable function. It takes as *input* a pair of numbers $(x, y) \in \mathbf{R}^2$, and produces as *output* a vector $\mathbf{v}(x, y) \in \mathbf{R}^2$. Figure 8.1.1 is a visualization of this vector-valued function, drawing at (x, y) a copy of the vector $\mathbf{v}(x, y)$ based at (x, y).

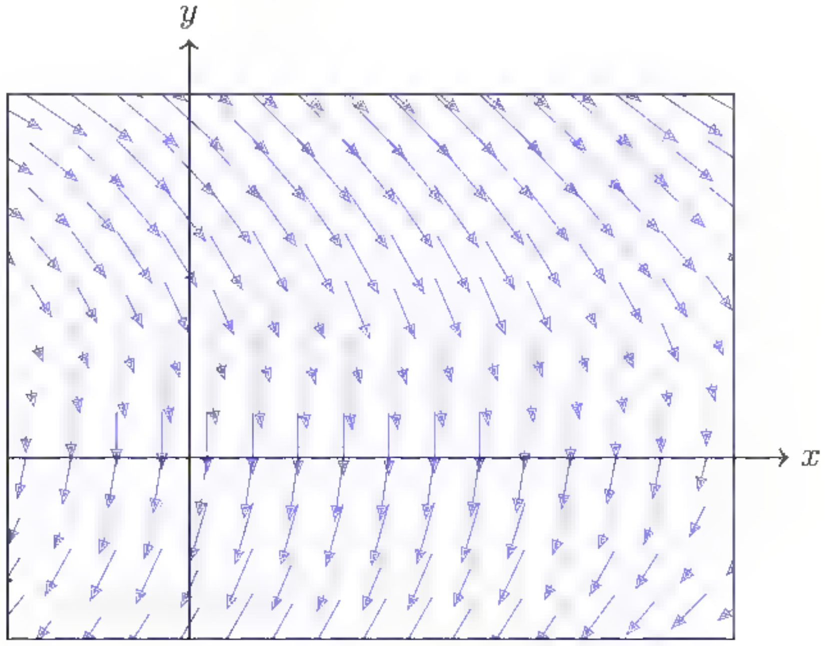


FIGURE 8.1.1. Wind velocity represented as a vector-valued function of (x, y).

Writing $\mathbf{v}(x, y) = (v_1(x, y), v_2(x, y))$ in terms of component functions, the value of v_2 is the "northerly component" of the wind direction, so when $v_2(a, b) < 0$ at some point (a, b) it means that the wind direction at (a, b) points in a southern (rather than northern) direction. ■

Example 8.1.10. In the previous example, the wind velocity at any given point will also change with time. So we may want to regard wind velocity as a function of position and time:

$$\mathbf{v}(x, y, t) = \text{wind velocity at the point } (x, y) \text{ at time } t.$$

This function takes as input a vector (x, y, t) in $\mathbf{R}^3$ and produces as output a vector $\mathbf{v}(x, y, t)$ in $\mathbf{R}^2$. One might try to visualize this as a movie version of Figure 8.1.1 in which the sea of arrows is undulating in time. But as the number of input variables increases further or the output becomes an m-vector for $m > 3$ we can't literally visualize it at all. Nonetheless, these low-dimensional visualizations can be a helpful psychological device when confronting many-variable functions.

If we are interested in atmospheric modeling, practically all the quantities of interest – wind speed, temperature, pressure, humidity – vary according to position and time, and therefore are given by multivariable functions. These multivariable functions are related by a complicated system of differential equations involving *lots* of variables. The difficulty of solving these differential equations is one reason that weather prediction is so difficult. ■

Example 8.1.11. Suppose a company uses 3 inputs to make 2 type of products, and the total amount of each of the 3 inputs is denoted as x_1, x_2, x_3 respectively. The amount of each of the two output products may then be considered as functions $p_1(x_1, x_2, x_3)$ and $p_2(x_1, x_2, x_3)$, so the vector-valued function

$$\mathbf{p} : \mathbf{R}^3 \to \mathbf{R}^2$$

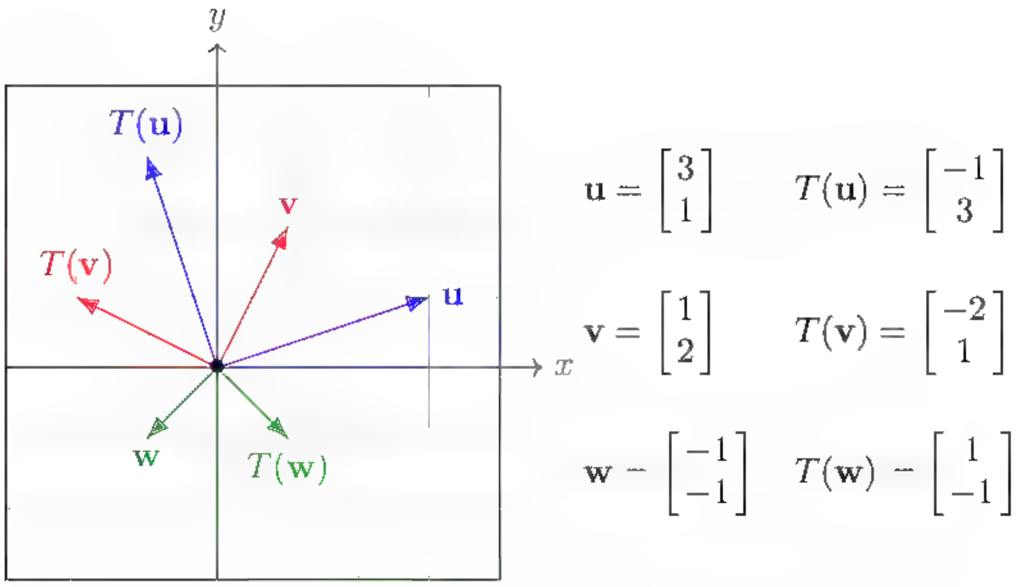
defined by $\mathbf{p}(x_1, x_2, x_3) = (p_1(x_1, x_2, x_3), p_2(x_1, x_2, x_3))$ keeps track of the company's entire production activity for these two products.

Such a mathematical model involves some idealization: we may be ignoring dependence of each p_i on information more refined than the raw total amount of each of the 3 inputs. Moreover, as usual in modeling of finance, we are treating the input amounts x_i as if they may vary continuously through all real-number values, whereas in reality they might vary only through some large but finite range of values (since prices do not come in increments smaller than some basic unit, such as 1 cent in the US economy, and there is only a finite amount of money in the world). The idealization of the x_i's as $\mathbf{R}$-valued variables allows us to use optimization techniques from calculus to analyze quantitative problems about the production process, and it works well in practical applications of economics. ■

Example 8.1.12. Vector-valued functions can encode ways to *manipulate vectors geometrically*. For example, the function $T : \mathbf{R}^2 \to \mathbf{R}^2$ given by

$$T\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} -y \\ x \end{bmatrix}$$

is a *rotation*.



y
$T(\mathbf{u})$
$\mathbf{v}$
$T(\mathbf{v})$
$\mathbf{u}$
x
$\mathbf{w}$
$T(\mathbf{w})$
$\mathbf{u} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$ $T(\mathbf{u}) = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$
$\mathbf{v} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ $T(\mathbf{v}) = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$
$\mathbf{w} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$ $T(\mathbf{w}) = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

FIGURE 8.1.2. Applying T to several vectors

What this means is that $T(\mathbf{v})$ gives you the vector $\mathbf{v}$ after rotating counterclockwise by 90° (when $\mathbf{v} \neq \mathbf{0}$). Figure 8.1.2 shows some specific input vectors to illustrate that T really is that rotation.

In a similar spirit, it turns out that the function $T : \mathbf{R}^2 \to \mathbf{R}^2$ given by

$$T\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} (\sqrt{3}/2)x + (1/2)y \\ -(1/2)x + (\sqrt{3}/2)y \end{bmatrix}$$

also rotates vectors. Can you figure out by how much T rotates vectors? Hint: the angle of rotation can be computed as the angle between $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $T\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right)$.

Later, in Section 14.4, we will study this type of example in more detail and give a complete answer for rotation through any angle. It leads to the study of matrices. ■

Example 8.1.13. For a given nonzero vector $\mathbf{v} \in \mathbf{R}^2$ and any vector $\mathbf{x} \in \mathbf{R}^2$, the projection

$$\mathbf{Proj}_{\mathbf{v}}(\mathbf{x}) = \left(\frac{\mathbf{x}\cdot\mathbf{v}}{\mathbf{v}\cdot\mathbf{v}}\right)\mathbf{v}$$

as in Section 6.2 is a function $\mathbf{R}^2 \to \mathbf{R}^2$ (where the vector $\mathbf{x}$ is the input). For example, if $\mathbf{v} = \begin{bmatrix}2\\4\end{bmatrix}$, then

$$\mathbf{Proj}_{\mathbf{v}}\left(\begin{bmatrix}x\\y\end{bmatrix}\right) = \left(\frac{2x+4y}{20}\right)\begin{bmatrix}2\\4\end{bmatrix} = \begin{bmatrix}(x+2y)/5\\(2x+4y)/5\end{bmatrix}.$$

This projection is illustrated in Figure 8.1.3 for a specific $\mathbf{x}$, but the visualization for this $\mathbf{v}$ is the same for any $\mathbf{x}$.

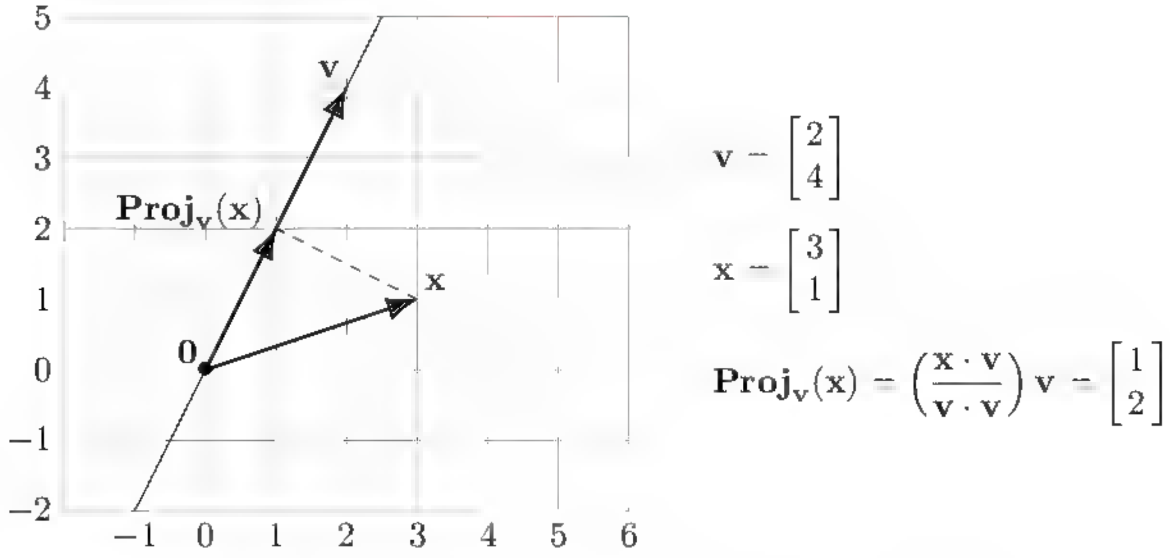

5
4
3
2
1
0
−1
−2
−1
0
1
2
3
4
5
6
v
Proj_v(x)
x
0
$\mathbf{v} = \begin{bmatrix}2\\4\end{bmatrix}$
$\mathbf{x} = \begin{bmatrix}3\\1\end{bmatrix}$
$\mathbf{Proj}_{\mathbf{v}}(\mathbf{x}) = \left(\frac{\mathbf{x}\cdot\mathbf{v}}{\mathbf{v}\cdot\mathbf{v}}\right)\mathbf{v} = \begin{bmatrix}1\\2\end{bmatrix}$

FIGURE 8.1.3. Projection of $\mathbf{x}$ onto the span of $\mathbf{v}$.

A related example that is ubiquitous in applications to computer graphics is the projection $T : \mathbf{R}^3 \to \mathbf{R}^3$ into the xz-plane (or some other plane), expressing how a 3-dimensional object (such as a ball or a house) is to be presented on a flat screen, as in Figure 8.1.4.

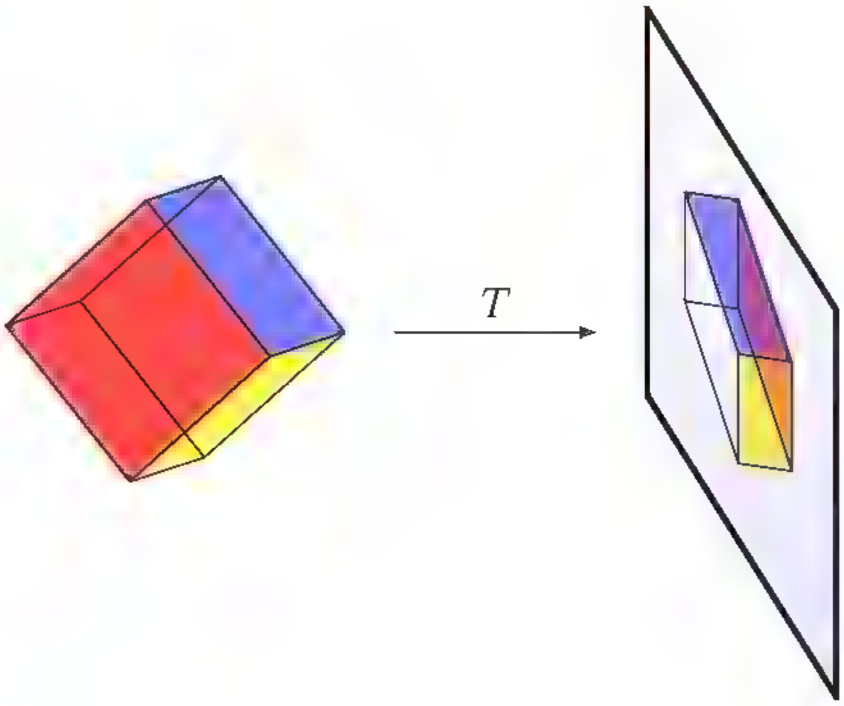


FIGURE 8.1.4. A 2-dimensional image of a 3-dimensional object

In other words, $T(\mathbf{v})$ is the point in the xz-plane where a point $\mathbf{v} \in \mathbf{R}^3$ is shown in a 2-dimensional rendering of a 3-dimensional object (and one may include shading or other data in the 2-dimensional image to provide a sense of depth, creating the illusion of 3 dimensions within a flat screen). ■

8.2. Composition. There is a way of making a new function from two old functions that you encountered when studying the Chain Rule in single-variable calculus: feeding one function into another.

Example 8.2.1. For functions $f : \mathbf{R} \to \mathbf{R}$ and $g : \mathbf{R} \to \mathbf{R}$, the *composition* $f \circ g : \mathbf{R} \to \mathbf{R}$ is defined by $(f \circ g)(x) = f(g(x))$. As an illustration, the function $h(x) = \sin(x^2)$ is the composition $f \circ g$ for $g(x) = x^2$ and $f(u) = \sin(u)$. Also, $E(x) = e^{x^3 - x}$ is the composition $f \circ g$ for $g(x) = x^3 - x$ and $f(u) = e^u$. Even something as mundane as raising a function to a power is a composition. Indeed, for any function $g : \mathbf{R} \to \mathbf{R}$, the function $g(x)^7$ is the composition $f \circ g$ for $f(u) = u^7$. ■

Beware that when composing functions, *the order of composition matters a lot*! For instance, if $f(x) = x + 1$ and $g(x) = 2x$, then

$$(f \circ g)(x) = 2x + 1,$$

whereas

$$(g \circ f)(x) = 2(x + 1) = 2x + 2.$$

For more complicated f and g, the distinction between $f \circ g$ and $g \circ f$ can be rather more dramatic. As an example, if $f(x) = x^2$ and $g(x) = e^x$ then $(f \circ g)(x) = f(e^x) = e^{2x}$ whereas $(g \circ f)(x) = g(x^2) = e^{x^2}$ (which grows much more rapidly than e^{2x} as $x \to \infty$).

Just as with functions $\mathbf{R} \to \mathbf{R}$, we can form the composition of vector-valued functions:

Definition 8.2.2. If $\mathbf{g} : \mathbf{R}^n \to \mathbf{R}^p$ and $\mathbf{f} : \mathbf{R}^p \to \mathbf{R}^m$ are multivariable functions (note that $\mathbf{g}$ has output belonging to $\mathbf{R}^p$ on which $\mathbf{f}$ is applied), we can form a new *composite* function:

take an input in $\mathbf{R}^n$; first apply $\mathbf{g}$ to it, and then apply $\mathbf{f}$:

$$\text{input } \mathbf{x} \in \mathbf{R}^n \overset{\mathbf{g}}{\rightsquigarrow} \mathbf{R}^p \overset{\mathbf{f}}{\rightsquigarrow} \mathbf{R}^m$$

As a shorthand, we write this new function as $\mathbf{f} \circ \mathbf{g}$; the symbol $\circ$ is read as "composed with." In symbols, the new function is given by

$$(\mathbf{f} \circ \mathbf{g})(\mathbf{x}) = (\mathbf{f} \text{ applied to } \mathbf{g}(\mathbf{x})) = \mathbf{f}(\mathbf{g}(\mathbf{x}))$$

Example 8.2.3. Consider the functions $\mathbf{f} : \mathbf{R}^2 \to \mathbf{R}^2$ and $\mathbf{g} : \mathbf{R}^2 \to \mathbf{R}^2$ defined by

$$\mathbf{f}(u, v) = (uv, u + v), \quad \mathbf{g}(x, y) = (e^{xy}, x - y)$$

(so $f_1(u, v) = uv$, $f_2(u, v) = u + v$, $g_1(x, y) = e^{xy}$, $g_2(x, y) = x - y$). Then $\mathbf{f} \circ \mathbf{g} : \mathbf{R}^2 \to \mathbf{R}^2$ evaluated at $(x, y) \in \mathbf{R}^2$ equals

$$(\mathbf{f} \circ \mathbf{g})(x, y) = \mathbf{f}(\mathbf{g}(x, y)) = \mathbf{f}(e^{xy}, x - y) = (e^{xy}(x - y), e^{xy} + x - y).$$

In this example, the composite function $\mathbf{g} \circ \mathbf{f} : \mathbf{R}^2 \to \mathbf{R}^2$ also makes sense. Its value on input $(u, v) \in \mathbf{R}^2$ is

$$(\mathbf{g} \circ \mathbf{f})(u, v) = \mathbf{g}(\mathbf{f}(u, v)) = \mathbf{g}(uv, u + v) = (e^{uv(u+v)}, uv - (u + v)) = (e^{u^2v + uv^2}, uv - u - v).$$

Observe that in this case, $\mathbf{f} \circ \mathbf{g}$ and $\mathbf{g} \circ \mathbf{f}$ are *very different* functions (just look at the formulas we have computed for each). **The order of composition matters** (familiar for scalar functions: $\sin(x^2) \neq \sin(x)^2$ as functions). ■

Example 8.2.4. What is the composition of the rotation function T (from Example 8.1.12) and the projection $P = \mathbf{Proj}_{\begin{bmatrix} 2 \\ 4 \end{bmatrix}}$ (from Example 8.1.13)? To work out $(T \circ P)\left(\begin{bmatrix} x \\ y \end{bmatrix}\right)$, we first apply P to $\begin{bmatrix} x \\ y \end{bmatrix}$, yielding $\begin{bmatrix} (x+2y)/5 \\ (2x+y)/5 \end{bmatrix}$, and then we apply T to this, yielding $\begin{bmatrix} -(2x+y)/5 \\ (x+2y)/5 \end{bmatrix}$. Therefore,

$$(T \circ P)\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} -(2x+y)/5 \\ (x+2y)/5 \end{bmatrix}.$$

Now compute $P \circ T$ (i.e., composition in the reverse order) and confirm it is *very different* from $T \circ P$. Once again: **order of composition matters**. Can you think about this visually without doing the algebra? ■

Example 8.2.5. Consider the function:

$$T : \mathbf{R}^2 \to \mathbf{R}^2,$$
$$T\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} -y \\ x \end{bmatrix}.$$

As illustrated in Example 8.1.12, this function has a nice geometric interpretation: T takes an input vector $\mathbf{v}$ and rotates it 90° counterclockwise around the origin to produce the output vector $T\mathbf{v}$.

For example,

$$T\left(\begin{bmatrix} 3 \\ 4 \end{bmatrix}\right) = \begin{bmatrix} -4 \\ 3 \end{bmatrix}$$

and

$$T\left(\begin{bmatrix} -4 \\ 3 \end{bmatrix}\right) = \begin{bmatrix} -3 \\ -4 \end{bmatrix},$$

so

$$(T \circ T)\left(\begin{bmatrix} 3 \\ 4 \end{bmatrix}\right) = T\left(T\left(\begin{bmatrix} 3 \\ 4 \end{bmatrix}\right)\right) = T\left(\begin{bmatrix} -4 \\ 3 \end{bmatrix}\right) = \begin{bmatrix} -3 \\ -4 \end{bmatrix}.$$

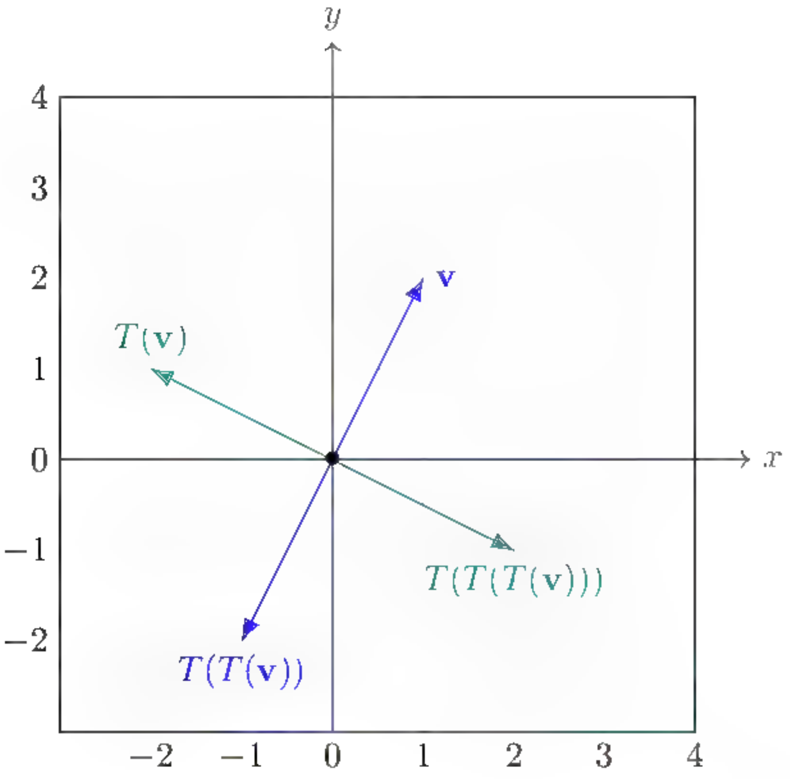
In general, the same calculation shows that

$$(T \circ T)\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} -x \\ -y \end{bmatrix} = -\begin{bmatrix} x \\ y \end{bmatrix}.$$

In other words,

$$(T \circ T)(\mathbf{v}) = -\mathbf{v}.$$

Geometrically, this makes sense: if you a rotate vector $\mathbf{v}$ by 90° counterclockwise around the origin twice, you end up with the vector $-\mathbf{v}$ of the same length but pointing in the opposite direction. This is illustrated in Figure 8.2.1, even with a third application of T.



$$\mathbf{v} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \qquad T(T(\mathbf{v})) = \begin{bmatrix} -1 \\ -2 \end{bmatrix}$$

$$T(\mathbf{v}) = \begin{bmatrix} -2 \\ 1 \end{bmatrix} \qquad T(T(T(\mathbf{v}))) = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

FIGURE 8.2.1. Repeatedly rotating by 90° counterclockwise returns in 4 steps

■

Example 8.2.6. For the functions $\mathbf{g} : \mathbf{R} \to \mathbf{R}^3$ and $\mathbf{f} : \mathbf{R}^3 \to \mathbf{R}^2$ defined by

$$\mathbf{g}(t) = (t, \cos(t), \sin(t)), \quad \mathbf{f}(x, y, z) = (y, z),$$

g can be visualized as the path of a particle moving on a helix on the cylinder $y^2 + z^2 = 1$ of radius 1 around the x-axis as shown in Figure 8.2.2, and f is the projection onto the yz-plane. Then $\mathbf{f} \circ \mathbf{g} : \mathbf{R} \to \mathbf{R}^2$ is given by

$$(\mathbf{f} \circ \mathbf{g})(t) = \mathbf{f}(t, \cos(t), \sin(t)) = (\cos(t), \sin(t)),$$

the path of the particle's "shadow" in the yz-plane moving counterclockwise around a red circle of radius 1 in the yz-plane shown in Figure 8.2.2.

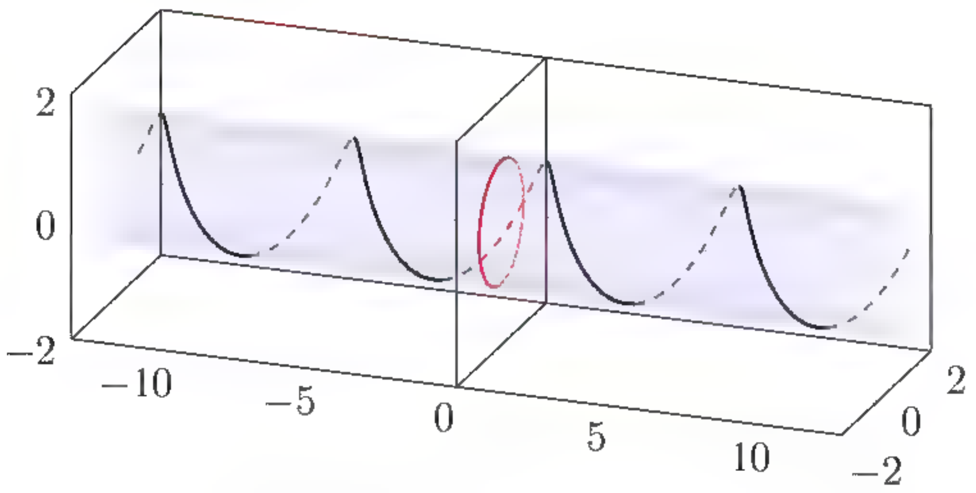


FIGURE 8.2.2. Particle moving in the positive x-direction on a helix, with shadow in the yz-plane ($x = 0$) tracing out a circular red path.

On the other hand, the composition $\mathbf{g} \circ \mathbf{f}$ the other way around *doesn't make any sense*! Indeed, f has output living in the yz-plane but the input for g is a scalar, so it makes no sense to evaluate g at a point $\mathbf{f}(x, y, z) \in \mathbf{R}^2$. ■

8.3. Graphs, level sets, and contour plots. When working with functions that involve several variables, one is immediately confronted with the question of how to think about them. For functions of one variable, $y = f(x)$, we have basically two ways: we can study the function computationally or algebraically by using a formula for $f(x)$, *or* we can study the geometry of its graph as a subset of the plane $\mathbf{R}^2$: this consists of the points (x, y) for which $y = f(x)$.

In our experience with functions $f(x)$ of one variable, it can be quite helpful to visualize the function graphically. We now describe two ways of doing this for a function $f : \mathbf{R}^n \to \mathbf{R}$ of n variables:

- using the graph of f (appropriately defined below),
- using the "level sets" of f (which are certain curves when $n = 2$) via diagrams to be called "contour plots" (analogues of topographic maps that show elevations on land to hikers and bathymetric maps that show depths near the ocean floor to submarines or other deep-sea vehicles).

Since a dominant theme in this course is to focus on techniques and viewpoints that are uniformly applicable to $\mathbf{R}^n$ for all n, we want to emphasize right away that *in real-world applications with n-variable functions for $n \geq 2$, the graph is often (though not always) less useful than the contour plot.* This may sound hard to believe, due to your experience with using graphs to think about everything in single-variable calculus, but it is true. We will come back to this in Section 8.4 after discussing the core definitions and some examples.

Much as the graph of a 1-variable function $f(x)$ is the subset of $\mathbf{R}^2$ defined as

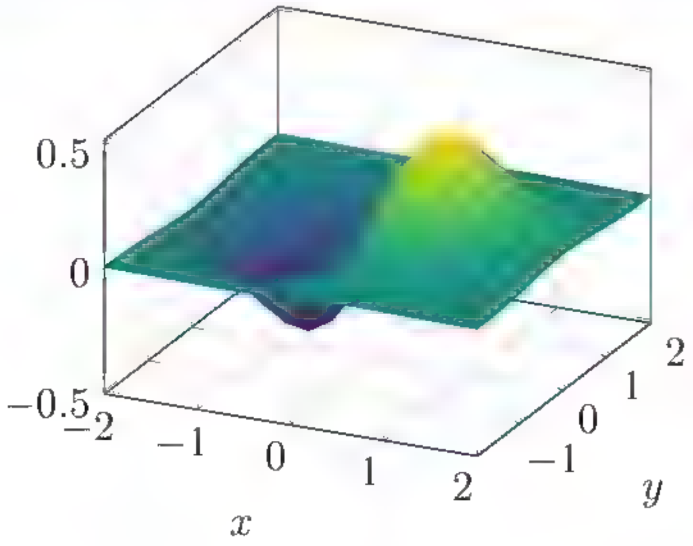
$$\text{Graph}(f) = \{(x, y) \in \mathbf{R}^2 : y = f(x)\},$$

for an n-variable function $f : \mathbf{R}^n \to \mathbf{R}$ its graph is a subset of $\mathbf{R}^{n+1}$ defined as follows:

Definition 8.3.1. The *graph* of $f : \mathbf{R}^n \to \mathbf{R}$ is the subset of $\mathbf{R}^{n+1}$ (not $\mathbf{R}^n$!) defined as

$$\text{Graph}(f) = \{(x_1, \cdots, x_n, z) \in \mathbf{R}^{n+1} : z = f(x_1, \cdots, x_n)\}.$$

For example,
the graph of $f(x, y) = x\exp(-x^2 - y^2)$
is the surface shown on the right.



The fact that the graph lies in $\mathbf{R}^{n+1}$ makes it impossible to literally be drawn when $n > 2$. For our purposes its main role will arise in the case $n = 2$, since the graph of a function $f : \mathbf{R}^2 \to \mathbf{R}$ is a subset of $\mathbf{R}^3$ that we (or better: a computer) can draw; *don't confuse this with the contour plot in* $\mathbf{R}^2$. Such graphs are a helpful way of taking our first steps into the world of multivariable functions, as we shall see.

Example 8.3.2. As a first illustration, let's work out the graph of the function $f(x, y) = \sqrt{1 - x^2 - y^2}$. We can only take the square root of a nonnegative number, so we require $1 - x^2 - y^2 \geq 0$, or equivalently

$$x^2 + y^2 \leq 1.$$

This is the disk D centered at the origin in $\mathbf{R}^2$ with radius 1, on the left in Figure 8.3.1.

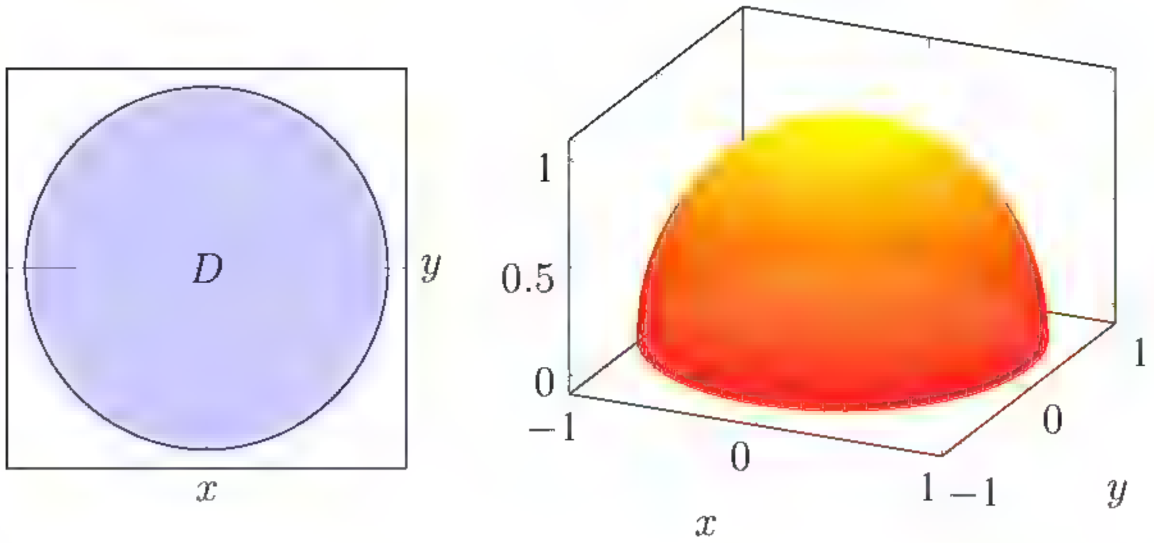


FIGURE 8.3.1. The disc D (in $\mathbf{R}^2$) and graph (in $\mathbf{R}^3$) of $f(x,y) = \sqrt{1-x^2-y^2}$ over D.

The graph of f is therefore

$$\begin{aligned}\mathrm{Graph}(f) &= \{(x,y,z) \in \mathbf{R}^3 \,:\, (x,y) \in D,\ \ z = f(x,y)\} \\ &= \{(x,y,\sqrt{1-x^2-y^2}) \,:\, (x,y) \in D\}\end{aligned}$$

Notice that since $z = \sqrt{1-x^2-y^2}$, we then have $x^2 + y^2 + z^2 = 1$ with $z \geq 0$. This graph is the upper hemisphere of the sphere in $\mathbf{R}^3$ with radius 1 centered at $(0,0,0)$, on the right in Figure 8.3.1. ■

Having seen graphs in $\mathbf{R}^3$ for some 2-variable functions, we now turn to a different (and ultimately more useful) way to get a feeling for the behavior of an n-variable function $f : \mathbf{R}^n \to \mathbf{R}$ for $n \geq 2$. We can try to visualize f by considering what are called its "level sets". For $n = 2$ this is something you may have encountered in real life (without the "level set" terminology) in the following context:

Example 8.3.3. If you are hiking in a park and you get a "contour map" (or "contour plot") of the terrain you are about to hike in, it may look something like Figure 8.3.2 below.

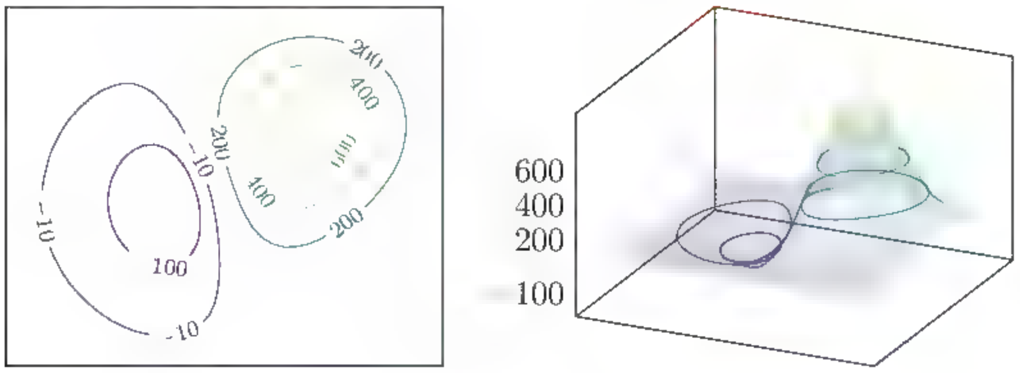


FIGURE 8.3.2. A contour plot and the graph of the altitude function $A : \mathbf{R}^2 \to \mathbf{R}$.

The curves on the left in Figure 8.3.2 indicate where the terrain is at a fixed level. For example, the curve that is labeled by 400 represents where the terrain has an altitude of 400 feet. The mathematical way to think about this is to consider the altitude function,

$$z = A(x,y).$$

The set of points (x,y) where $A(x,y) = 400$ is the curve on the contour map labeled 400. The set of points (x,y) where $A(x,y) = 600$ is the curve on the contour map labeled 600. In general, the set of points (x,y) where $A(x,y) = c$ is called the *level curve* of the function A at level c (and is also called a *level set*, or sometimes even a *contour line* even though it generally looks nothing at all like a line).

The contour map consisting of a collection of level curves is very helpful in visualizing the altitude function $z = A(x, y)$, and gives us a good understanding of the terrain. Of course, the contour map doesn't show the level curves $A(x, y) = c$ for *every* c; that would be impossible. Rather, the contour map shows these level curves for "enough" values of c that one can get a sense of the hilliness of the terrain for practical purposes. Between level curves drawn for values $c_1 < c_2$ are level curves for intermediate values of c that are omitted for clarity.

The c's also can be encoded as colors with varying intensity, yielding a contour plot that is a "heat map" as for temperatures in Figure 8.3.3 (see this website for the current version of the same). ■

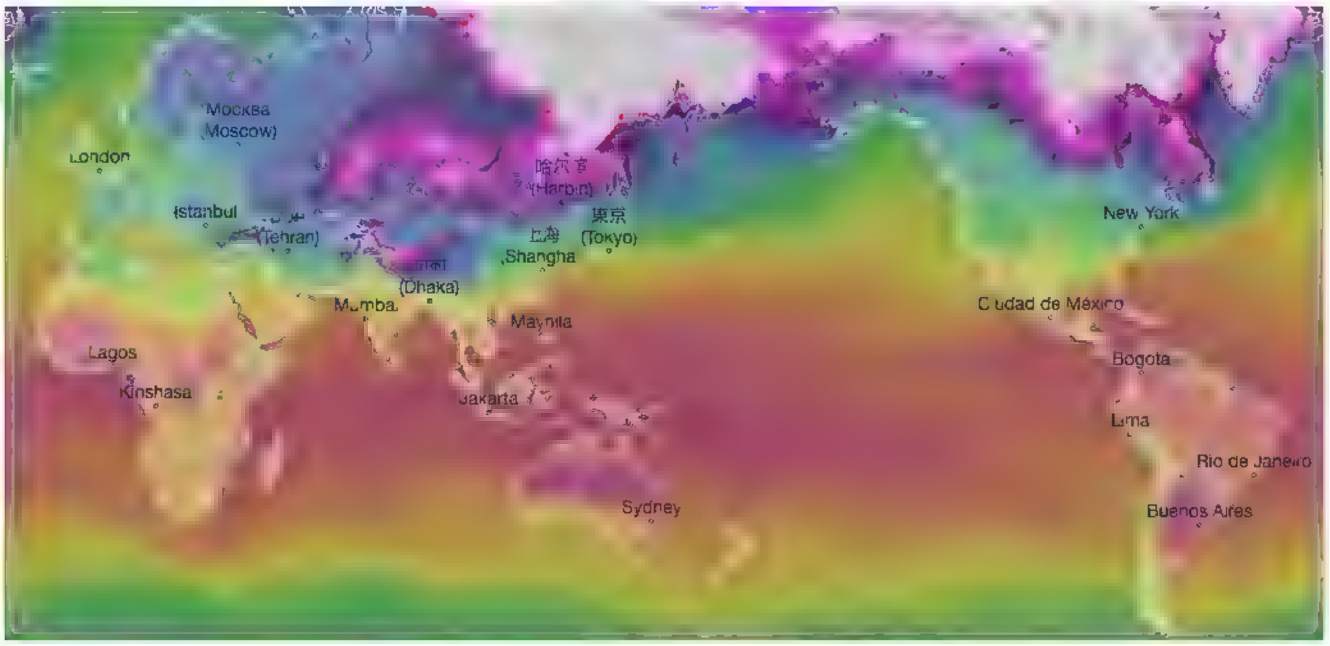


FIGURE 8.3.3. A "heat map" serving as a contour plot for the temperature everywhere.

Adapting the preceding idea to functions of any number of variables goes as follows:

Definition 8.3.4. Let $f : \mathbf{R}^n \to \mathbf{R}$ be a function. For any $c \in \mathbf{R}$, the *level set of f at level c* is the set of points $(x_1, \cdots, x_n) \in \mathbf{R}^n$ for which $f(x_1, \cdots, x_n) = c$.

If f is a function $\mathbf{R}^2 \to \mathbf{R}$ of 2 variables then a *contour plot* of f is a picture in $\mathbf{R}^2$ that depicts the level sets of f for many different values of c (often values with some common difference for "consecutive" level sets, such as a common difference of 10, or 4, or 1, or 0.2, etc).

Example 8.3.5. Consider a function $g(x, y)$ whose contour plot is as shown in Figure 8.3.4. Looking at the values of g on the various level sets in this region, we observe that the values are changing in a definite direction as we approach the each of the points P and Q.

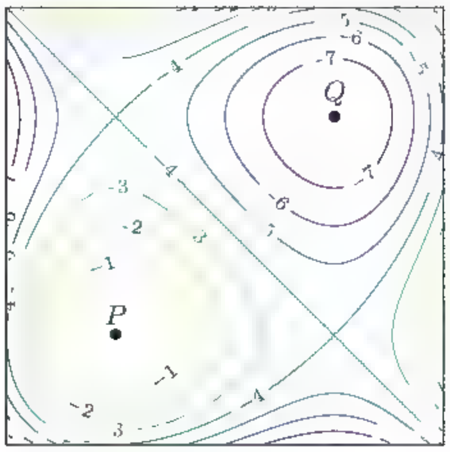


FIGURE 8.3.4. Contour plot for g with approximate local extrema marked

This contour plot suggests that g attains a local maximum at P with $g(P) \approx 0$ and a local minimum at Q with $g(Q) \approx -8$. Even without seeing the surface graph of g in $\mathbf{R}^3$, the contour plot in $\mathbf{R}^2$ indicates special geometry near the local extrema. There are also self-crossing level sets, making approximate "X" shapes; this indicates a different type of behavior for g, as we shall understand later. ■

Remark 8.3.6 (online resource). The website Desmos plots level curves $g(x, y) = c$ for varying c, and the command ContourPlot in Wolfram makes the contour plot of $g(x, y)$ for x and y varying between chosen bounds (a sample is here). To explore surface graphs, try CalcPlot3D or GeoGebra.

The main lesson from this discussion is that for general n and a function $f : \mathbf{R}^n \to \mathbf{R}$, there are two quite different notions that can be used to explore the behavior of f (such as to find "local extrema" of f, a topic you explored in great depth for $n = 1$ in single-variable calculus and which we will study in multiples ways for general n in the rest of the course): its *graph* as a subset of $\mathbf{R}^{n+1}$, and its collection of level sets $\{\mathbf{x} \in \mathbf{R}^n : f(\mathbf{x}) = c\}$ that are certain subsets of $\mathbf{R}^n$ (e.g., for $n = 2$ these are typically curves, which assemble for enough values of c to provide a contour plot for f on flat paper or computer screen).

8.4. Why contour plots? In Chapters 9 and 10 (for multivariable first derivatives) as well as in Chapter 26 (for multivariable second derivatives), we will use contour plots to explain and learn about multivariable derivatives and optimization problems for functions $f : \mathbf{R}^n \to \mathbf{R}$ with $n \geq 2$. Throughout that entire discussion, the graphs of such functions f will play *no essential role whatsoever* (even in the case $n = 2$, which will be the primary case for worked examples and for which we *could* think about the graph as a surface in $\mathbf{R}^3$ that *will* sometimes be drawn – for motivational purposes only).

Given the ubiquity of modern software that allows one to create and manipulate 3-dimensional images on a computer screen, for the study of functions of 2 variables (as a segue into the case of n variables for general $n \geq 2$) you might wonder why there is an emphasis on 2-dimensional contour plots rather than on 3-dimensional graphs. There are several reasons for this:

(i) (mathematical) Fundamental optimization techniques such as gradient descent (Section 11.3) and Lagrange multipliers (Chapter 12) that work for functions $f : \mathbf{R}^n \to \mathbf{R}$ with *any* n take place in $\mathbf{R}^n$ and *not* on the graph in $\mathbf{R}^{n+1}$. For $n = 2$, this requires thinking about a function $f(x, y)$ from the perspective of $\mathbf{R}^2$, and that is exactly what the skill of working with the 2-dimensional contour plot (rather than the 3-dimensional graph) is based upon.

(ii) (pedagogical) Thinking too much about graphs can cause confusion about the mechanism that underlies gradient descent (an optimization technique to be discussed in Section 11.3): "the gradient vector points in the direction where the function increases most rapidly" (Theorem 11.3.2). At a point (a, b), the *gradient* of $f(x, y)$ (a type of vector-valued multivariable first derivative of f that we will define) is a vector in $\mathbf{R}^2$ that can be visualized in terms of a contour plot, but students who focus on graphs often erroneously think the gradient is a vector in $\mathbf{R}^3$ pointing up along the graph surface $z = f(x, y)$ (at the point $(a, b, f(a, b))$); this is totally wrong.

(iii) (scientific) Among practicing scientists (in physics, neuroscience, economics, biology, chemistry, statistics, etc.), experimental outcomes continue to be *more often* communicated via 2-dimensional contour plots than via 3-dimensional graphs. This may be the case because it is generally easier to convey quantitative information (gradients, constraints, descent methods, etc.) in a contour plot. Also, neuroscience studies show rather convincingly that most people are better at understanding 2-dimensional images than 3-dimensional images, no matter how nice the graphics (possibly because our brain receives information about the visual world as a 2-dimensional projection onto the retina, and reconstructs depth late in the visual system).

Certainly 3-dimensional images serve a useful role (such as when displaying the geometry of a molecule or protein), but 3-dimensional *graphs* of 2-dimensional data can be hard to interpret without the ability to rotate them (such as when giving a typical slide talk or writing a scientific paper).

Remark 8.4.1. The need to fundamentally change one's way of thinking when generalizing calculus concepts beyond the 1-dimensional case is not limited to the role of graphs. For example, in single-variable calculus you computed a lot of definite integrals $\int_a^b f(x)dx$. Though you have encountered these to compute areas and then some volumes (and arc-lengths), the utility of the definite integral goes much further, including tasks in statistics, economics, and natural sciences that involve no geometry at all. What *really* unifies these disparate applications of the definite integral is the need to "add up the contributions of many small parts". It is this latter viewpoint that is the best way to think about what definite integration means, why it shows up everywhere, and how the idea of definite integration adapts to the multivariable setting.

Chapter 8 highlights (links to highlights in and next chapters)

Notation	**Meaning**	**Location in text**
$\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$	a function with input in $\mathbf{R}^n$ and output in $\mathbf{R}^m$	start of Section 8.1 (and Table 0.0.1)
$\mathbf{f} \circ \mathbf{g}$	for $\mathbf{g} : \mathbf{R}^n \to \mathbf{R}^p$ and $\mathbf{f} : \mathbf{R}^p \to \mathbf{R}^m$, it the function $\mathbf{R}^n \to \mathbf{R}^m$ defined by $(\mathbf{f} \circ \mathbf{g})(\mathbf{x}) = \mathbf{f}(\mathbf{g}(\mathbf{x}))$	Definition 8.2.2

Concept	**Meaning**	**Location in text**
scalar-valued function	a function with input in $\mathbf{R}^n$ and output in $\mathbf{R}$	Definition 8.1.1
vector-valued function	a function with input in $\mathbf{R}^n$ and output in $\mathbf{R}^m$ (any m)	Definition 8.1.7
component function (or coordinate function)	for vector-valued $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$, it is any $f_j : \mathbf{R}^n \to \mathbf{R}$ where $f_j(\mathbf{x})$ is jth entry in $\mathbf{f}(\mathbf{x})$ $(1 \le j \le m)$	Definition 8.1.7
composite function	for $\mathbf{g} : \mathbf{R}^n \to \mathbf{R}^p$ whose output is input to $\mathbf{f} : \mathbf{R}^p \to \mathbf{R}^m$, it is $\mathbf{f} \circ \mathbf{g} : \mathbf{R}^n \to \mathbf{R}^m$ defined by "first apply $\mathbf{g}$, then apply $\mathbf{f}$"	Definition 8.2.2
graph of $f : \mathbf{R}^n \to \mathbf{R}$	the collection of points $(x_1, \dots, x_n, f(x_1, \dots, x_n))$ in $\mathbf{R}^{n+1}$	Definition 8.3.1
level set of $f : \mathbf{R}^n \to \mathbf{R}$ (at level c)	the collection of points $\mathbf{x} \in \mathbf{R}^n$ satisfying $f(\mathbf{x}) = c$	Definition 8.3.4
contour plot (in $\mathbf{R}^2$)	picture in $\mathbf{R}^2$ of many level curves of $f : \mathbf{R}^2 \to \mathbf{R}$ labeled by level values (usually changing by a fixed increment); **not** to be confused with surface graph in $\mathbf{R}^3$ (see Figure 8.3.2)	Definition 8.3.4

Result	**Meaning**	**Location in text**
order of composition (when both make sense) matters	for $\mathbf{g} : \mathbf{R}^n \to \mathbf{R}^m$ and $\mathbf{f} : \mathbf{R}^m \to \mathbf{R}^n$, usually $\mathbf{f} \circ \mathbf{g} : \mathbf{R}^n \to \mathbf{R}^n$ and $\mathbf{g} \circ \mathbf{f} : \mathbf{R}^m \to \mathbf{R}^m$ are very different (even when $n = m$)	Example 8.2.3, Example 8.2.4

Skill	**Location in text**
for $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$, read off its scalar-valued component functions $f_1, \dots, f_m$	illustrated in Example 8.2.3
understand that rotations and projections are vector-valued functions	Example 8.1.12, Example 8.1.13, Figure 8.1.4
compute composition $\mathbf{f} \circ \mathbf{g}$ of given $\mathbf{g} : \mathbf{R}^n \to \mathbf{R}^p$ and $\mathbf{f} : \mathbf{R}^p \to \mathbf{R}^m$ by either working algebraically with component functions or interpreting geometrically (with rotations and projections)	Examples 8.2.3–8.2.6
recognize that composition of functions in some order may make no sense	end of Example 8.2.6
for $f : \mathbf{R}^2 \to \mathbf{R}$, relate basic information in contour plot (in $\mathbf{R}^2$) and surface graph (in $\mathbf{R}^3$)	Example 8.3.3
identify directions of increase and decrease for $f : \mathbf{R}^2 \to \mathbf{R}$ from labels in its contour plot	Example 8.3.5

9. Partial derivatives and contour plots

With the experience from Chapter 8 under our belts, and our knowledge of some linear algebra, we are ready to begin the study of multivariable calculus. Given a scalar-valued function $f(x_1, \dots, x_n)$ of n variables $x_1, \dots, x_n$, we will be able to differentiate it "with respect to" each variable x_i. These are called *partial derivatives*; we shall introduce them, explain them graphically, and discuss what they have to do with finding maxima or minima.

By the end of this chapter, given a scalar-valued function $f : \mathbf{R}^n \to \mathbf{R}$ (see Table 0.0.1 for the function notation $f : A \to B$) you should be able to:

- compute symbolically the partial derivatives of f, as well as second partial derivatives;
- recognize and interpret, from a graph or contour plot of f, positive and negative partial derivatives.

9.1. Single-variable derivative review. One of the cornerstones of single-variable calculus is the derivative of a function. In this section we review its definition and discuss some of its interpretations, focusing on a few aspects that reappear in the more general setting of functions of n variables.

Single-variable calculus is the study of functions $f : \mathbf{R} \to \mathbf{R}$, or $f : I \to \mathbf{R}$ for some interval I in $\mathbf{R}$. There are many examples of such functions:

Example 9.1.1. If you are taking a car trip and t represents time (measured in minutes) since the journey began, then $f(t)$ could be the total distance you have traveled between times 0 and t. ■

Example 9.1.2. As an example from economics, x can represent the total amount of goods produced in a factory and $C(x)$ the total cost (say in dollars) of this production. ■

Example 9.1.3. In biology, $P(t)$ could measure the total population of fish in a fixed environment, such as a lake, as a function of time t. Typically one is interested in how this function depends on various external factors such as the size of the environment, the number and type of predators, mortality rates, etc. (and that dependence may impose conditions on the possibilities for P, sometimes expressed in terms of a differential equation). ■

Remark 9.1.4. In the second example above, both the independent variable x and the value taken by the function C are discrete rather than continuous: we do not measure fractions of a cent. However, it is useful to treat such discrete variables as if they vary continuously so that we can study them using methods from calculus. "Mathematical modeling" of discrete phenomena using continuous methods is a common technique that works extremely well in practice, though one must always keep in mind that it is just a model.

In your study of calculus you have encountered many different functions $\mathbf{R} \to \mathbf{R}$, and are familiar with how to associate to such a function its graph. For simplicity, we adopt a uniform notation and always call the independent (input) variable x and the dependent (output) variable y (i.e., $y = f(x)$) and draw the graph of the function in the xy-plane.

The derivative of a function $f(x)$ at a point $x = c$ is written in one of the equivalent forms

$$f'(c), \quad \frac{df}{dx}(c), \quad \left.\frac{df}{dx}\right|_{x=c}. \tag{9.1.1}$$

Here are some ways you may have seen this quantity described:

(i) A basic interpretation of the derivative is that it represents the "instantaneous rate of change". If we change the value of x slightly from $x = c$ to something nearby, say $c + 10^{-6}$, then the derivative provides a good approximation for how much $f(x)$ changes. Indeed, $f'(c)$ is (approximately) the *ratio* of the change of output to the (small) change in input near $x = c$. So if we increase from c to $c + 10^{-6}$, then to a good degree of approximation, the value of f has changed from $f(c)$ to $f(c) + f'(c)10^{-6}$.

The derivative $f'(c)$ multiplies the change of x, here 10^{-6}, to compute the difference between the two values of $f(x)$. There is nothing special about 10^{-6}; we could consider any small change from $x = c$ to $x = c + h$, where h represents the tiny increment in the independent variable (with smaller h making $f'(c)$ an even better approximation to the *ratio of changes* of output to input: $f(c+h) \approx f(c) + f'(c)h$, or equivalently $f(x) \approx f(c) + f'(c)(x - c)$ for x near c).

(ii) The number $f'(c)$ is the *slope* of the line tangent to the graph of f at the point $(c, f(c))$. Recall that if we consider all possible lines $y = mx + b$, then only some of these pass through the point $(c, f(c))$. This point lies on such a line precisely when $f(c) = mc + b$, so this tells us what b is if we know m (namely, $b = f(c) - mc$), so to determine the line we just need to know its slope.

If we consider all lines that pass through this point, then we expect that precisely one will have a slope making it a "best fit" with the graph of f near this point. Strictly speaking, this is only true if f is differentiable at $x = c$. (For instance, the graph of the function $f(x) = |x|$ has no best-fitting line through $(0, 0)$.) Assuming that f is differentiable, once we have shown that there is *only one* best-fitting line (in a suitable sense) we define $f'(c)$ to be its slope m.

We have been discussing the derivative at a specific point $x = c$, but it is convenient to think of the derivative as a function too (by varying c). Thus we write

$$f'(x) \text{ or } \frac{df}{dx}$$

for the function which assigns to every point x the value $f'(x)$, the derivative of f at x. (This assumes that f has a derivative at every point x.)

We have exhibited two rather different ways of thinking about derivatives: (i) they describe the sensitivity in the values of f to small changes in the independent variable ($f(c + h) \approx f(c) + f'(c)h$ for h near 0), and (ii) they are purely geometric quantities (slopes of tangent lines). Like so many ideas in mathematics, derivatives can be interpreted in many different ways.

Remark 9.1.5. When we consider functions which represent quantities arising in some part of science or engineering, then their derivatives sometimes have names specific from those disciplines. These may provide further motivation or intuition.

In Example 9.1.1 where $f(t)$ is the distance traveled, $f'(t)$ is the *instantaneous velocity* at time t. For the total cost of production $C(x)$ in Example 9.1.2, one calls $C'(x)$ the *marginal cost*: how much it costs to produce 1 extra unit of product (informally: $C(x + 1) \approx C(x) + C'(x) \cdot 1$). Finally, in Example 9.1.3, whether $P'(t)$ is positive or negative tells us whether the population is increasing or decreasing at that moment of time, and $|P'(t)|$ indicates how sharply the population growth or decline is at that time.

Regardless of these interpretations, we emphasize that the primary mathematical definition of the derivative is that $f'(c)$ is the limit of difference quotients:

$$f'(c) = \lim_{h \to 0} \frac{f(c+h) - f(c)}{h}.$$

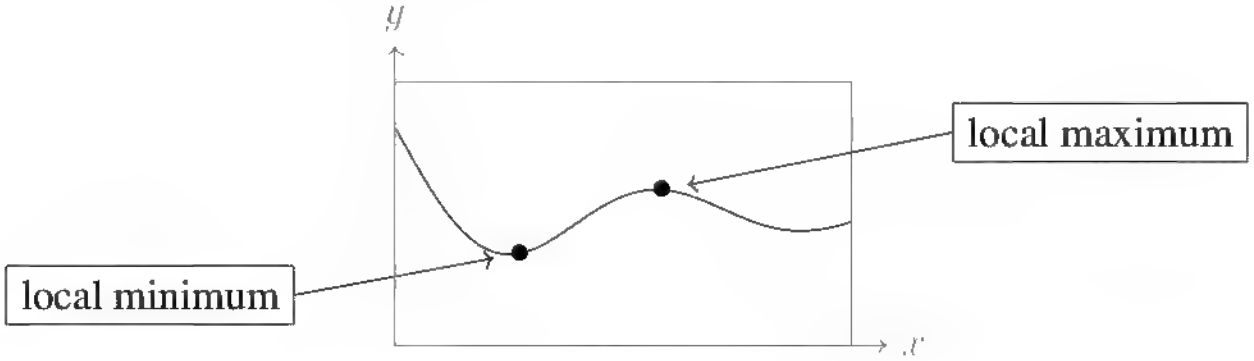
This is what the derivative "really is." Depending on how the function is presented to us, this definition may be hard to work with directly. For relatively simple functions given by expressions involving polynomials, exponentials, etc., you have learned how to compute $f'(x)$, again as a function involving the same types of expressions, without needing to go back to the original limit definition.

It is easy to get drawn into thinking that these formulas for $f'(x)$ for familiar functions $f(x)$ are all that derivatives are, but they are not! They are simply the outcome of specific calculations which are ultimately deduced from the limit definition. Such formulas are very convenient and useful in many situations, but they apply only when $f(x)$ is given by some simple expression. If you know the function $f(x)$ only through a sampling of its values, such as when x is the mathematical model for a quantity that is really discrete, then such formulas are not of much help at all.

We now turn to some of the uses of the derivative. The primary one is to describe basic features of the graph $y = f(x)$. In particular, this graph has 'peaks' and 'valleys', and the derivative $f'(x)$ helps us to understand these.

Definition 9.1.6. A point $x = c$ is called a *local minimum* for f if, for values of x close to c, $f(x)$ is at least as big as $f(c)$; i.e., $f(x) \geq f(c)$ at least when x is fairly close to c. We picture this by thinking of the point $(c, f(c))$ as being at the bottom of a "well".

We define a *local maximum* (at $x = c$) similarly: $f(x) \leq f(c)$ for all x near c.



A useful fact from single-variable calculus is that if $f(x)$ is differentiable at every x, and if $x = c$ is a local maximum or local minimum for f, then $f'(c) = 0$. In other words, at such points the tangent line is horizontal, or alternatively (in terms of our other interpretation of the derivative) the value of $f(x)$ is insensitive (to first order) to small changes in x.

So to find the local maxima or local minima of $f(x)$, we can proceed by first computing the function $f'(x)$ and then solving $f'(x) = 0$ (and hope the "second derivative test" applies there). Neither of these steps may be simple. For example, if f is a polynomial of degree 25 then it is straightforward (if tedious) to calculate $f'(x)$ as a polynomial of degree 24, but it is probably very difficult or impossible to find exact solutions to $f'(x) = 0$. On the other hand, if we only know the values of $f(x)$ through discrete sampling, as can happen when f arises in certain mathematical models, then we will not be able to compute $f'(x)$ exactly but at best only find some reasonable approximation to it.

However, finding the solutions x of $f'(x) = 0$ is not the whole story! For instance:

Example 9.1.7. Suppose that we have carried out this step for some function f, and have obtained a list of numbers, say $x = 3, 6.5, 12$ and 137, where the derivative vanishes. We must then decide whether each of these values is a local maximum, a local minimum, or perhaps neither.

There is a helpful test involving the second derivative of $f(x)$, which we will review in Example 26.1.1, but even that is *sometimes inconclusive* (consider $f(x) = x^3$ and $f(x) = \pm x^4$). To repeat: finding where $f'(x) = 0$ is only a preliminary step, since it merely indicates the values of x which we need to study more closely when seeking local maxima or local minima. ■

Keep in mind that there are points at which $f'(x) = 0$ that are neither local maxima nor local minima, such as $f(x) = x^3$. At such critical points, the graph is concave-up on one side and concave-down on the other side:

Example 9.1.8. Consider $f(x) = x^3$: this has derivative vanishing only at $x = 0$, but $f(x)$ is negative for $x < 0$ and positive for $x > 0$, so $x = 0$ is neither a local maximum or minimum for f. The graph of $y = x^3$ is concave-down when $x < 0$ and concave-up when $x > 0$, as shown in Figure 9.1.1.

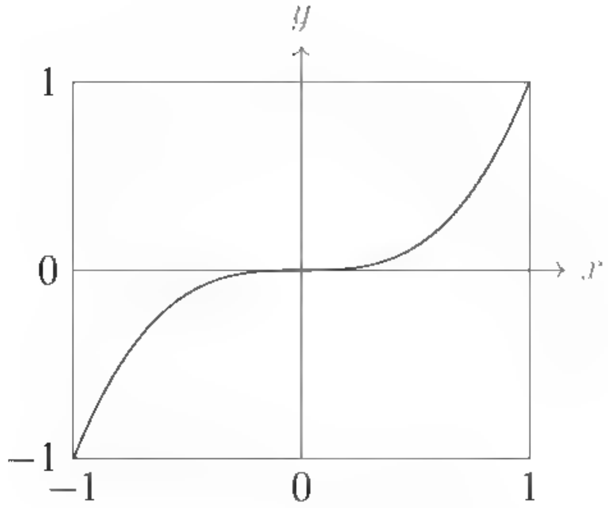


FIGURE 9.1.1. The graph of $f(x) = x^3$ over the interval $[-1, 1]$.

■

This concludes our review of derivatives of functions of one variable.

9.2. Partial derivatives, a first look. Consider a function $f(x_1, x_2)$ of 2 variables. Just as for functions of one variable, we can ask: "What is the instantaneous rate of change of f at some point $(x_1, x_2) = (a, b)$?" This pervades multivariable optimization in economics, natural sciences, etc., as will be seen later on. Unlike the single-variable case discussed in Section 9.1, where there is essentially just one meaning (though there is a variety of ways to interpret it), with more than one independent variable there are multiple possible meanings. "Partial" derivatives will provide a first step in this direction, but these and a few other ideas to be introduced later (directional derivatives, the gradient, and the derivative matrix) will ultimately all fit together into a coherent picture that tells us how to best approximate $f(x_1, x_2)$ near (a, b) with a "linear" function.

The main idea for "partial" derivatives is to consider the function $f(x_1, x_2)$ as a collection of functions of one variable, which we can do in *two* different ways. Firstly, for each value $x_2 = b$, the function $f(x_1, b)$ is a function of one variable, and as we choose different values of b, we get different functions of x_1.

Example 9.2.1. Suppose $f(x_1, x_2) = \sin(x_1 + x_2^2)$. At $x_2 = 1$ the resulting function $f(x_1, 1)$ of one variable is $\sin(x_1 + 1)$. Likewise at $x_2 = 1/2$ it is $f(x_1, 1/2) = \sin(x_1 + 1/4)$, at $x_2 = 0$ it is $f(x_1, 0) = \sin x_1$, and so on.

On the other hand, we could equally well think of x_1 as being equal to some fixed value a, and then consider the function of one variable $f(a, x_2)$. For $f(x_1, x_2) = \sin(x_1 + x_2^2)$, when $x_1 = 0$ we get $\sin(x_2^2)$, for $x_1 = 1$ we get $\sin(1 + x_2^2)$, and so on. ■

Let's interpret this visually via the *graph* of f in $\mathbf{R}^3$: the set of points $(x_1, x_2, f(x_1, x_2))$. It is a surface in space whose height over (x_1, x_2) is $f(x_1, x_2)$ (see Figure 9.2.1). If we set x_2 equal to some value, say $x_2 = 3$, then we are considering the function $f(x_1, 3)$; the graph of this function of one variable is where the graph of f meets the vertical plane $x_2 = 3$. If we set x_2 equal to another value, say $x_2 = 4$, this corresponds to sliding the vertical plane over to $x_2 = 4$, which meets the graph of f in a different curve (see Figure 9.2.1). Thus we can think of the surface graph of f as a collection of curves, each obtained where the surface meets a vertical plane $x_2 = b$; see Figure 9.2.1. This is a way of reducing the situation to something we have studied before: algebraically we go from the function of two variables $f(x_1, x_2)$ to a collection of functions $f(x_1, b)$ of one variable, to which we can apply what we know from single-variable calculus. Geometrically, we think of the graph of f as the collection of graphs of single-variable functions.

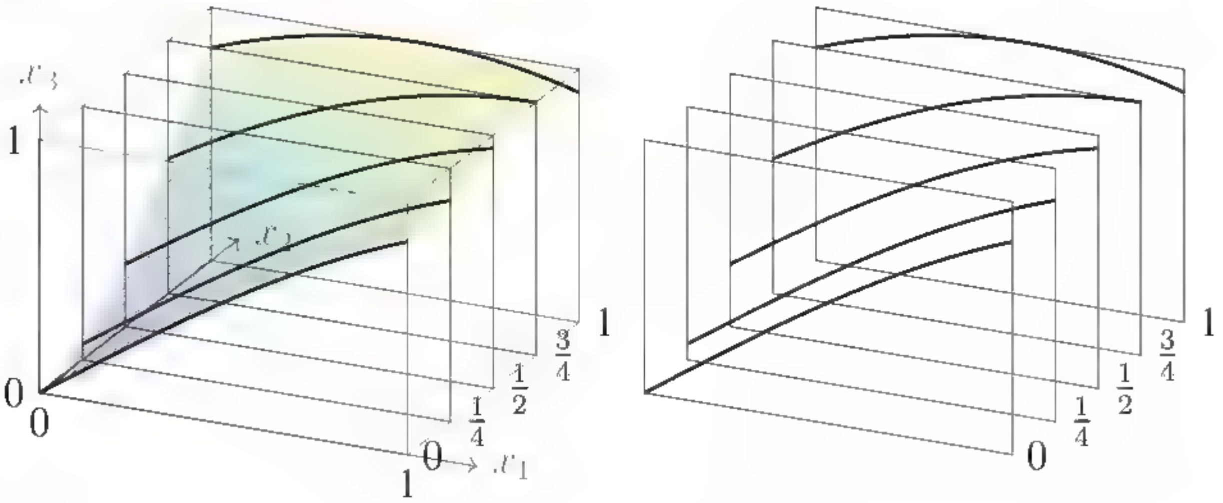


FIGURE 9.2.1. The graph of $f(x_1, x_2)$ and its slices by the planes $x_2 = 0$, $x_2 = 1/4$, $x_2 = 1/2$, $x_2 = 3/4$, $x_2 = 1$ as the respective graphs of $f(x_1, 0)$, $f(x_1, 1/4)$, $f(x_1, 1/2)$, $f(x_1, 3/4)$, $f(x_1, 1)$.

Slicing the surface graph with the planes $x_1 = a$ in another direction could have been considered instead. That would correspond to working with the function $f(a, x_2)$ for various a.

We are now ready to define partial derivatives. Let (a, b) be any point. Suppose we hold x_2 equal to a fixed value b. As we just discussed, we can think of this as corresponding to the intersection of the graph of f with the plane $x_2 = b$, or symbolically as the function of one variable $f(x_1, b)$.

Definition 9.2.2. The *partial derivative of f with respect to x_1 at the point (a, b)*, denoted in any of the equivalent ways

$$\frac{\partial f}{\partial x_1}(a,b), \quad \left.\frac{\partial f}{\partial x_1}\right|_{(a,b)}, \quad f_{x_1}(a,b),$$

means the derivative of the function $f(x_1, b)$ at $x_1 = a$.

Said differently, the partial derivative of f with respect to x_1 at (a, b) is the instantaneous rate of change of f at the point (a, b) if we *only move in the x_1-direction* (so x_2 is held constant, at the value b)! The formal definition is once again as a limit of difference quotients,

$$\frac{\partial f}{\partial x_1}(a,b) = \lim_{h \to 0} \frac{f(a+h,b) - f(a,b)}{h}. \tag{9.2.1}$$

Observe that in (9.2.1) we only vary the first coordinate of the input and leave the second coordinate at b. We repeat for emphasis: $\partial f/\partial x_1$ entails keeping x_2 constant.

In precisely the same way, we can define the *partial derivative of f with respect to x_2 at (a, b)*, denoted

$$\frac{\partial f}{\partial x_2}(a,b), \quad \left.\frac{\partial f}{\partial x_2}\right|_{(a,b)}, \quad f_{x_2}(a,b).$$

As above, the actual definition is the limit of difference quotients

$$\frac{\partial f}{\partial x_2}(a,b) = \lim_{h \to 0} \frac{f(a,b+h) - f(a,b)}{h}, \tag{9.2.2}$$

now varying the x_2-coordinate and holding x_1 fixed at the value a.

Example 9.2.3. Consider $f(x_1, x_2) = x_1^2 + x_2^2$. To compute its partial derivative with respect to x_1 at $(3, 2)$, we first set $x_2 = 2$; i.e., consider the function $f(x_1, 2) = x_1^2 + (2)^2 = x_1^2 + 4$. The partial derivative value $f_{x_1}(3,2)$ is just the derivative of this function at $x_1 = 3$. Thus we compute $f_{x_1}(x_1, 2) = 2x_1$ and hence $f_{x_1}(3,2) = 6$.

We could have done this a bit more directly. Namely, without first explicitly setting $x_2 = 2$, we differentiate the two summands in $x_1^2 + x_2^2$ separately. The derivative of x_1^2 with respect to x_1 is $2x_1$, while the derivative of x_2^2 with respect to x_1 is 0 because x_2^2 does not change at all if we vary only x_1. Thus we obtain that $f_{x_1}(x_1, x_2) = 2x_1$ for any (x_1, x_2), so $f_{x_1}(3,2) = 6$. ■

Just as for functions of one variable, we can think of partial derivatives in either of the two directions as *new functions*. Thus $\partial f/\partial x_1$ (or f_{x_1}) is the instantaneous rate of change of f in the x_1-direction at any point (x_1, x_2), and $\partial f/\partial x_2$ is the instantaneous rate of change of f in the x_2-direction at any point (x_1, x_2). (The technique of "implicit differentiation" from single-variable calculus is understood better via partial derivatives, but the link involves the multivariable Chain Rule that is discussed in Chapter 17.) For functions of two variables there are two instantaneous rates of change: one in each of the two coordinate directions. For functions of three (or n) variables, there are corresponding instantaneous rates of change (i.e., partial derivatives), one with respect to each independent variable, as we shall see in Section 9.3.

In the remainder of this chapter we take up two questions:

(i) How do we actually compute partial derivatives symbolically for multivariable functions such as $\sin(x_1 + x_2^2)$ or $e^{x_1^3 x_2}$ which are built from familiar functions such as polynomials, trigonometric functions, exponential functions, etc.?

(ii) Even without having a simple formula for the function f, how can we interpret partial derivatives in terms of visual data attached to f, at least in the 2-variable case?

It is a pleasant surprise that the partial derivatives in the coordinate directions can be used to build a more fundamental concept of "total derivative" that enables us to to compute a notion of partial derivative *in any direction at all*. This involves further linear algebra (via matrices), so we postpone it to Chapter 13.

9.3. Partial derivatives, symbolically. We now practice computing partial derivatives of functions built from familiar expressions involving polynomials, exponentials, and trigonometric functions.

Example 9.3.1. Consider the function $f(x_1, x_2) = \sin(x_1^2 x_2^3)$. We now compute its two partial derivatives at the point $(-3, 2)$, first treating the partial derivative with respect to x_1.

Method 1 (symbolic): First, as a function of (x_1, x_2),

$$\frac{\partial f}{\partial x_1}(x_1, x_2) = 2x_1 x_2^3 \cos(x_1^2 x_2^3). \tag{9.3.1}$$

To obtain this we have applied the same principle as in the second half of Example 9.2.3: think of this as a function of x_1 alone, and regard x_2 as a *fixed constant* (so we are varying only x_1). We then apply the Chain Rule in the usual way: we differentiate as a function of x_1 with x_2 treated as a fixed numerical quantity, getting $\dfrac{\partial f}{\partial x_1}(x_1, x_2) = \cos(x_1^2 x_2^3)\dfrac{\partial}{\partial x_1}(x_1^2 x_2^3) = \cos(x_1^2 x_2^3)(2x_1 x_2^3)$, getting (9.3.1). Now evaluate at $(-3, 2)$ to get $f_{x_1}(-3, 2) = -(6)(8)\cos((9)(8)) = -48\cos 72$ (where the trigonometric functions are evaluated on input in radians).

Method 2 (numerical): Rather than first working symbolically throughout (treating x_2 like a fixed numerical quantity whose actual value we don't specify until we set it to be 2 after the x_1-differentiation), we could have *first* set $x_2 = 2$ before computing any derivatives. Since $f(x_1, 2) = \sin(8x_1^2)$, differentiating with respect to x_1 yields $f_{x_1}(x_1, 2) = 16x_1\cos(8x_1^2)$, so finally $f_{x_1}(-3, 2) = -48\cos 72$, as before.

Likewise, to compute $f_{x_2}(-3, 2)$ we can proceed symbolically (Method 1) to get

$$\frac{\partial f}{\partial x_2} = 3x_1^2 x_2^2 \cos(x_1^2 x_2^3),$$

so $f_{x_2}(-3,2) = 108\cos(72)$. If we instead proceed numerically (Method 2), we set $x_1 = -3$ to arrive at the function $f(-3, x_2) = \sin(9x_2^3)$ of x_2 and differentiate that to obtain $27x_2^2\cos(9x_2^3)$. Evaluating this at $x_2 = 2$ gives $f_{x_2}(-3,2) = 27(2)^2\cos(9\cdot 2^3) = 108\cos(72)$. ■

Example 9.3.2. Let's compute the partial derivatives of $g(x_1, x_2) = e^{4x_1 - 5x_2}$ at $(1, -1)$.

First we compute the partial derivative with respect to x_1. Using Method 2 as in Example 9.3.1, we substitute $x_2 = -1$ to obtain $g(x_1, -1) = e^{4x_1+5}$, so differentiating gives $g_{x_1}(x_1, -1) = 4e^{4x_1+5}$ and finally substituting $x_1 = 1$ yields $g_{x_1}(1, -1) = 4e^9$.

Alternatively, following Method 1 as in Example 9.3.1, we treat x_2 as if it is a fixed constant but we do not specify which constant value it has. This gives $g_{x_1}(x_1, x_2) = 4e^{4x_1 - 5x_2}$, so evaluating at $(1, -1)$ yields $g_{x_1}(1, -1) = 4e^{4+5} = 4e^9$ once again.

To compute the partial derivative with respect to x_2, we first use the numerical method (i.e., Method 2 from Example 9.3.1), so we substitute $x_1 = 1$ to get the function $g(1, x_2) = e^{4-5x_2}$ of x_2, which has derivative $-5e^{4-5x_2}$. Evaluating this in turn at $x_2 = -1$ gives that $g_{x_2}(1, -1) = -5e^9$.

Alternatively, using Method 1 as in Example 9.3.1, treat x_1 as a constant but do not specify which constant value it has. This gives $g_{x_2}(x_1, x_2) = -5e^{4x_1 - 5x_2}$, so $g_{x_2}(1, -1) = -5e^9$ once again. ■

We now phrase these two viewpoints (symbolic and numerical) for general functions.

If $f(x_1, \dots, x_n)$ is a function of n variables, then at the point $(a_1, \dots, a_n)$ its ith partial derivative, or equivalently its partial derivative with respect to x_i for any $i = 1, \dots, n$, denoted as

$$f_{x_i}(a_1, \dots, a_n) \text{ or } \frac{\partial f}{\partial x_i}(a_1, \dots, a_n) \text{ or } \left.\frac{\partial f}{\partial x_i}\right|_{(a_1, \dots, a_n)},$$

can be computed in two ways (in practice one always uses Method 1 below, but its equality with Method 2 below explains why partial derivatives satisfy some features of single-variable derivatives).

Method 1 (symbolic): think of the x_j's for $j \neq i$ as constant (*without* specifying their values) and apply the familiar single-variable differentiation rules in terms of x_i to obtain the new function of n variables, $f_{x_i}(x_1, \dots, x_n)$. Then substitute $(a_1, \dots, a_n)$ for $(x_1, \dots, x_n)$ throughout.

Method 2 (numerical): replace x_j with a_j in f for every $j \neq i$, and then differentiate the resulting function $f(a_1, \dots, a_{i-1}, x_i, a_{i+1}, \dots, a_n)$ of one variable x_i using the rules from *single-variable* calculus. Finally, *after that*, set $x_i = a_i$. In particular, for each i, the operation $\partial/\partial x_i$ satisfies the product and quotient rules (e.g., $\dfrac{\partial}{\partial x_i}(fg) = \dfrac{\partial f}{\partial x_i}g + f\dfrac{\partial g}{\partial x_i}$ for all f and g)!

Why mention two methods which look almost identical? The point is that Method 1 explains why Method 2 always proceeds "the same way" whatever values we use for the a_j's with $j \neq i$. In practice one uses the symbolic method, but thinking in terms of the numerical method illuminates more vividly how partial derivatives are related to single-variable derivatives.

9.4. Summary. Since this is such a fundamental part of the subject, we take the opportunity to briefly summarize what we have discussed above:

- For a single variable function, the usual derivative provides information about the change in values of a function when we make a small increase in the independent variable:

$$g(x+h) - g(x) \approx \frac{dg}{dx} \times h, \text{ for } h \text{ small.}$$

- For a function $f(x_1, x_2)$ of two variables, the partial derivative $\dfrac{\partial f}{\partial x_1}$ gives information about how the values of f change when we make a small increase in x_1 *while holding x_2 constant.*

  As an example, let's see what happens to $f(x_1, x_2) = x_1^2 + x_1 x_2^2$ when we go from $(x_1, x_2) = (2, 1)$ to $(x_1, x_2) = (2.1, 1)$; we have increased x_1 slightly, but hold x_2 fixed. We have:

$$\underbrace{f(2.1, 1) - f(2, 1)}_{=0.51} \approx \frac{\partial f}{\partial x_1}(2, 1) \times (0.1) = 0.5. \tag{9.4.1}$$

Make sure you understand (9.4.1). We will explain it visually in Section 9.5. To summarize:

- Let $f(x_1, x_2)$ be a scalar-valued function of (x_1, x_2). The partial derivative

  $$\frac{\partial f}{\partial x_1}, \text{ sometimes abbreviated as } f_{x_1},$$

  is the derivative of f with respect to x_1, holding x_2 as a constant. Similarly, $\frac{\partial f}{\partial x_2}$ – sometimes abbreviated as f_{x_2} – is the derivative of f with respect to x_2, holding x_1 as a constant.
- $\frac{\partial f}{\partial x_1}$ and $\frac{\partial f}{\partial x_2}$ are each scalar-valued functions of (x_1, x_2).
- Interpretation: if we hold x_2 constant and change x_1 by a small amount h, then approximately

  $$(\text{change in } f) = f(x_1 + h, x_2) - f(x_1, x_2) \approx \frac{\partial f}{\partial x_1} \times h. \tag{9.4.2}$$

  Similarly, if we hold x_1 constant and change x_2 by a small amount h, then approximately

  $$(\text{change in } f) = f(x_1, x_2 + h) - f(x_1, x_2) \approx \frac{\partial f}{\partial x_2} \times h. \tag{9.4.3}$$

- These concepts generalize directly if f depends on more variables. Let $f(x_1, \ldots, x_n)$ be a scalar-valued function of n variables. The *partial derivative*

  $$\frac{\partial f}{\partial x_i}, \text{ sometimes abbreviated as } f_{x_i},$$

  is the derivative of f with respect to x_i, regarding all other variables $x_1, x_2, \ldots, x_{i-1}$ and $x_{i+1}, x_{i+2}, \ldots, x_n$ as constants.

Remark 9.4.1 (optional). Beware that equations (9.4.2) and (9.4.3) are only approximations. Later, in Chapter 25, we will see how to improve this approximation by using second derivatives. In more advanced applications, it is sometimes important to understand precisely *how good* the approximation is. The first step toward this is the limit definition. For example, the mathematically precise formulation of (9.4.2) is

$$\lim_{h \to 0} \frac{f(x_1 + h, x_2) - f(x_1, x_2)}{h} = \frac{\partial f}{\partial x_1}(x_1, x_2).$$

Stated more plainly, the error in equation (9.4.2) is very small *relative to* h, when h is very small.

The functions we encounter in this course are "differentiable", which means that at every point the limit defining the "derivative" (in the sense of partial derivatives for now, but later we will develop a more powerful perspective) does indeed exist.

Just as for functions of one variable, there are however lots of functions that are not differentiable at some points. The simplest example is the function $f(x) = |x|$, which is not differentiable at $x = 0$. Likewise in the multivariable setting, the function $f(x_1, x_2) = |x_1 + x_2|$ does not admit a partial derivative f_{x_1} or f_{x_2} at $(0, 0)$ (that is, the limit defining such partial derivatives at $(0, 0)$ does not exist), and similarly at all points on the line where $x_1 + x_2 = 0$.

The reason we steer away from non-differentiable functions in this course is not that they are not useful (in fact, non-differentiable functions *do* arise in applications) but rather that their study requires different ideas and techniques. For those who are curious, here are some real-world contexts where non-differentiable functions play an essential role:

(i) (Brownian motion) The chaotic-looking behavior of particulate matter such as pollen or dust when suspended in a fluid was first observed empirically by the botanist Robert Brown in the early 19th century. The development of a mathematical theory of "Brownian motion" (for objects moving under the influence of random forces) began in the late 19th century, with applications by Einstein in 1905 to the kinetic theory of heat and gases (and in fact to the initial determination of the size of atoms and the related measurement of Avogadro's constant). In this mathematical theory, the jittery path followed by a particle under Brownian motion is modeled by a function of time that is continuous but (typically) *nowhere* differentiable!

(ii) (stochastic calculus) In many fields there are systems involving a lot of random noise and many interacting parts. Examples include finance (to predict future price of a stock, and the variance thereof), climate modeling, filtering theory in signal processing (used by computers on Apollo 11 during its descent to the Moon), and gene expression in biology. The mathematical framework for such problems is called stochastic calculus, in which many of the functions of time that arise are typically highly non-differentiable (as in Brownian motion).

Part of stochastic calculus is the study of differential equations involving "noise terms" (to model random forces in a system). A famous example of a stochastic differential equation is the *Black–Scholes equation* used to price certain financial instruments; the 1997 Nobel Prize in Economics was awarded for the discovery and study of this equation.

9.5. Partial derivatives, graphically. In Section 8.3, two tools in the study of functions f of 2 variables were discussed: the graph of f in $\mathbf{R}^3$ (the set of points (x, y, z) for which $z = f(x, y)$) and a contour plot of f in $\mathbf{R}^2$ (a collection of curves in the xy-plane). Our goal in this section is to understand what partial derivatives tell us about the contour plot. In practice contour plots are made by a computer, and Wolfram [illegible] is good at this; we won't ask you to create them on any exam, nor on homework except perhaps some basic examples whose level curves are things such as lines, circles, or parabolas.

Here is a contour plot (showing level sets at values in increments of 0.2) for a function $F(x, y)$ which may represent some experimental measurements (i.e., we may have *no explicit formula* for $F(x, y)$):

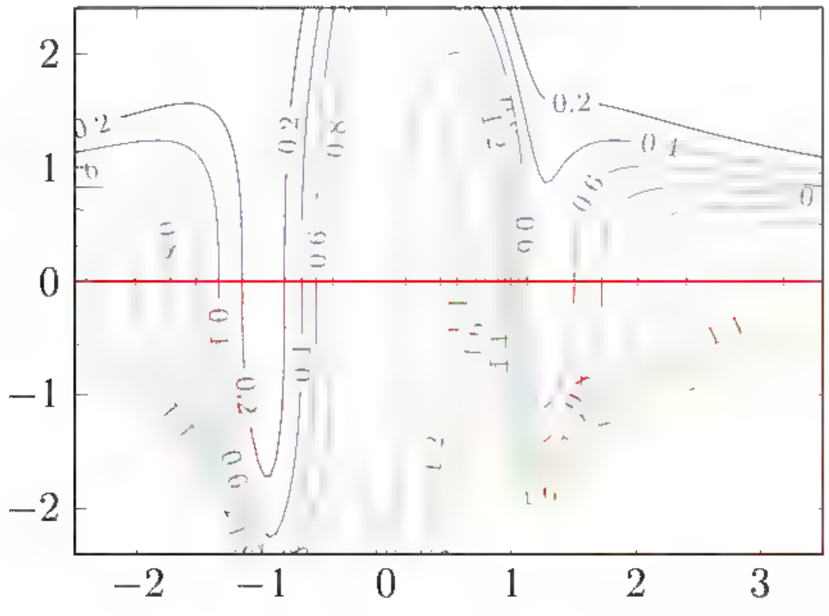


FIGURE 9.5.1. A contour plot for some $F(x, y)$ using increments of 0.2 in F-values

This contour plot is in the xy-plane, and the label on each curve indicates the F-value on that curve (in increments of 0.2). The x-axis extends from west to east and the y-axis extends from south to north. This contour plot is rather complicated, which indicates that the function $F(x, y)$ itself is fairly complicated.

We now use the 2-dimensional contour plot to read off qualitative information about the function $F(x, y)$. (Chapters 10–11 contain further discussion of the utility of 2-dimensional contour plots to express information about the behavior of 2-variable functions.)

Imagine that you are walking from west to east along the red line $y = 0$ in Figure 9.5.1 while on the surface graph $z = F(x, y)$. Let us analyze what you observe as you do this. The contour plot is a bird's eye view of the graph $z = F(x, y)$: your path looks like a straight line from the bird's perspective, but as you walk along the line $y = 0$ the function $f(x) = F(x, 0)$ on this line varies in its values and the graph of $f(x)$ as shown in Figure 9.5.2 keeps track of elevation as one walks along the red line.

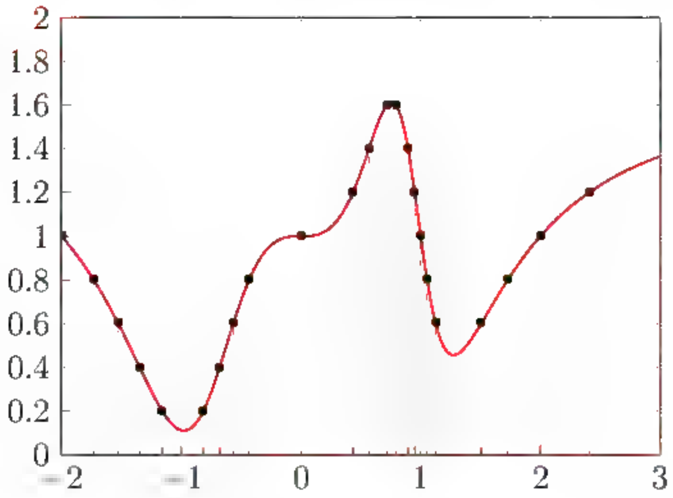


FIGURE 9.5.2. Graph of $f(x) = F(x, 0)$, marking increments of 0.2 vertically (F-values). The red curve is the slice of the surface $z = F(x, y)$ by the plane $y = 0$, the horizontal coordinate is x, and the vertical coordinate is z.

In Figure 9.5.2 we have drawn a dot on the red curve every time the altitude changes by 0.2 units. This value 0.2 agrees with how we have labeled the contour plot in Figure 9.5.1: the different contour curves in Figure 9.5.1 correspond to level sets for F at different values 0.2 units apart. We label each such dot with the altitude, and also drop a dashed vertical line down to the x-axis.

In other words, in Figure 9.5.2 there is one dot and an accompanying dashed line for each time you meet one of the contours in the contour plot. Note that the contour curves for $F(x, y)$ in Figure 9.5.1 are very close together near the point $(x, y) = (1, 0)$. In Figure 9.5.2, this corresponds to the fact that the x-coordinates of the dots are more tightly clustered near $x = 1$. In physical terms, elevation changes rapidly in this region since the plot shows level sets in *uniform* increments (0.2 in this case). By definition the derivative $f'(x)$ of $f(x) = F(x, 0)$ is the partial derivative $\frac{\partial F}{\partial x}$ at $(x, 0)$. Here are some observations:

- The regions where $f'(x) = \frac{\partial F}{\partial x}(x, 0)$ is positive are those where you are walking *uphill* in Figure 9.5.2 as you walk from west to east (i.e., it is where the numbers $f(x) = F(x, 0)$ on the labels *increase*). For example, along the interval of x-values in $[-1, 0]$ (with $y = 0$) is all uphill.
- The regions where $f'(x) = \frac{\partial F}{\partial x}(x, 0)$ is negative are those where you are walking *downhill*, again as you walk from west to east in Figure 9.5.2. The region along $[-2, -1]$ is downhill.
- When the absolute value $|f'| = |\frac{\partial F}{\partial x}|$ is large, the path is very steep. For example, it seems dangerously steep at $x = 1$. Notice that the contours (shown at *uniform* increments, 0.2 in this case) *are closer together* near $x = 1$. On the other hand, when the slope is more *gradual* (i.e., $|f'| = |\frac{\partial F}{\partial x}|$ is near 0), the contour curves (shown at *uniform* increments) *are further apart*. For instance, the slope is more gradual near $x = 2$.

- At points where $f' = \frac{\partial F}{\partial x} = 0$, the path momentarily *flattens out* (in the x-direction): this occurs here in the valley near $x = -1$ and on the peak to the left of $x = 1$, and also presumably in the flat area at $x = 0$.

More generally, for any b, $\frac{\partial F}{\partial x}(x, b)$ tells us about the slope of the path as we walk from west to east along the surface graph $z = F(x, y)$ on the curve in the surface lying above the horizontal line $y = b$, and the magnitude (i.e., absolute value) of this partial derivative at $x = a$ indicates the frequency with which we pass across contour lines as we move along the line $y = b$ near (a, b). High frequency of crossing contour lines (shown at uniform increments of F-values!) corresponds to $|F_x(a, b)|$ being large, and low frequency corresponds to $F_x(a, b) \approx 0$. We summarize all of this of follows:

Interpretation of partial derivatives on a contour plot: Visualize $F(x, y)$ as the height above (x, y) on the surface graph $z = F(x, y)$, where x is the east-west coordinate and y is the north-south coordinate (so larger values of x mean that we are further to the east, while larger values of y mean that we are further to the north). Then:

- $\frac{\partial F}{\partial x}(a, b)$ equals the slope (instantaneous change in altitude) experienced by someone walking on the surface $z = F(x, y)$ just as they go past the point (a, b) from *west to east*.
- $\frac{\partial F}{\partial y}(a, b)$ equals the slope (instantaneous change in altitude) experienced by someone walking on the surface $z = F(x, y)$ as they walk past the point (a, b) from *south to north*.

As an example, if $\frac{\partial F}{\partial y}(0, 0) > 0$ then a northerly path on the surface $z = F(x, y)$ is going *uphill* as it passes over $(0, 0)$.

On a contour plot of the function $F(x, y)$, the partial derivatives can be interpreted as follows:

- The *sign* of $\frac{\partial F}{\partial x}(a, b)$ tells us whether the labels of the contours (which represent the values of F) are *increasing* or *decreasing* as we walk through (a, b) from west to east.
- The *sign* of $\frac{\partial F}{\partial y}(a, b)$ tells us whether the values of F on the contours are *increasing* or *decreasing* as we walk through (a, b) from from south to north.
- If $\frac{\partial F}{\partial x}(a_1, b_1) > \frac{\partial F}{\partial x}(a_2, b_2) > 0$ then in the x-direction the slope at (a_1, b_1) is *steeper* than the slope at (a_2, b_2), so the contours (when shown for *uniform* increments in F-values) are spaced *closer together* as we move east across (a_1, b_1) than they are as we move east across (a_2, b_2). There is a corresponding statement for negative x-partial derivatives (still moving east). The situation for $\frac{\partial F}{\partial y}$ can be described in very similar terms (for moving north).

Example 9.5.1. Let's do a similar analysis for a path as shown in Figure 9.5.3 that heads north along the red line $x = 2.25$ on the same contour plot as in Figure 9.5.1.

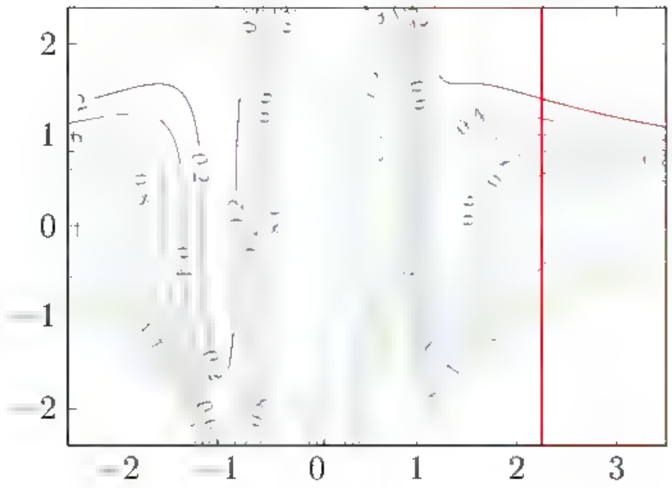


FIGURE 9.5.3. The line $x = 2.25$ (in red) and the same contour plot for $F(x, y)$

We want to address two issues concerning $F(2.25, y)$ as we walk up along the red line in Figure 9.5.3:

(i) Is the path uphill or downhill, or are there regions for each?
(ii) Is the path steeper at the beginning or the end of the path?

In each case, we want to interpret the answer in terms of properties of $\frac{\partial F}{\partial y}$.

When going upwards along the red line (so from south to north), the labels of the contours (level curves) are *decreasing*. For example they go down from 1.8 to 1.0, then to 0.6, and so on: the successive contours we cross have labels that decrease successively by 0.2. This tells us that the path is downhill, which says that the partial derivative $\frac{\partial F}{\partial y}$ is *negative* at all points where the red line intersects the contours.

Furthermore the distance between the contours starts out small, then is fairly large, and then gets smaller again. This tells us that the path is steepest at the beginning of the path (the furthest south), then levels out, and then gets a bit steeper again when we get further north. In terms of partial derivatives, this says that $\frac{\partial F}{\partial y}$ is most negative at the southern-most point where the red line intersects the contours, then gets larger (closer to zero) toward the middle of the path, and finally becomes more negative again as we proceed further north. ■

Example 9.5.2. Figure 9.5.4 is a contour plot of a function $f(x, y) = ax + by + c$, for certain (nonzero) constants a, b, and c, with contours shown for f-values in increments of 3.

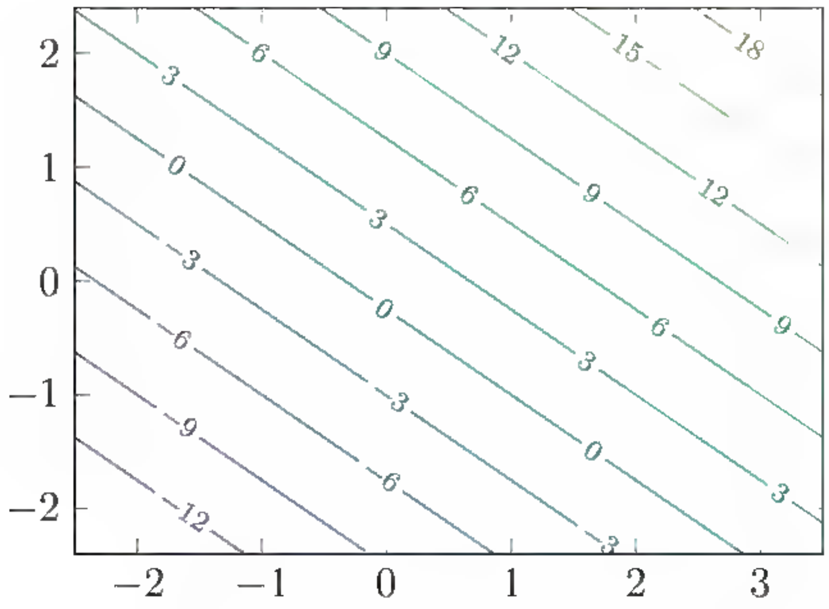


FIGURE 9.5.4. A contour plot corresponding to a surface graph that is a plane

Evaluating the first partial derivatives yields

$$\frac{\partial f}{\partial x}(x, y) = a \quad \text{and} \quad \frac{\partial f}{\partial y}(x, y) = b$$

This means for each unit distance we walk east, our altitude changes by a units (it rises if $a > 0$ and falls if $a < 0$, or stays the same if $a = 0$). And for each unit distance we walk north, our altitude changes by b units. The fact that these partial derivatives are constants – i.e., they do not depend on x and y – is expressed by the contour plot consisting of evenly spaced lines.

■

Example 9.5.3. In Figure 9.5.5 we give a contour plot of $f(x, y) = x(y^2 + 1)$ (using f-value increments of 1, but omitting odd f-value labels on the right side for readability) with the following points marked:

$$P = (0, 0) \qquad Q = (3/2, 0) \qquad R = (7/2, 0) \qquad S = (2, 1) \qquad T = (2, -1).$$

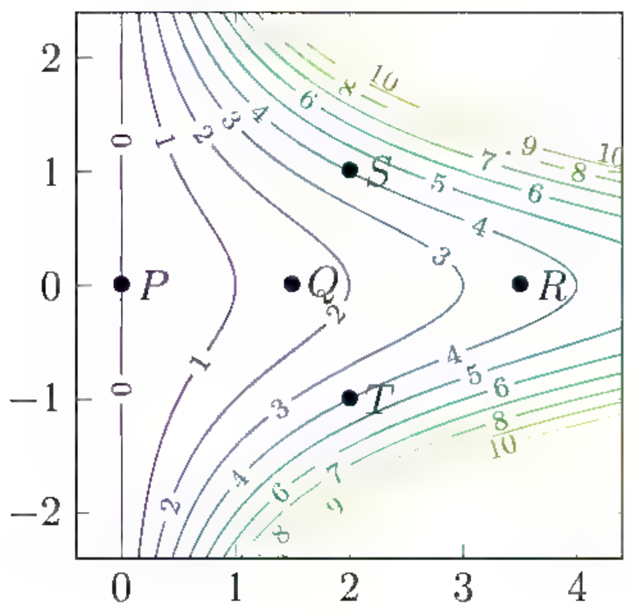


FIGURE 9.5.5. A contour plot for $f(x, y) = x(y^2 + 1)$ using increments of 1 in f-values

Let's determine the answers to the following two qualitative questions at each point, first by looking at the contour plot (if that gives enough information) and then by computing the partial derivatives of $f(x, y)$ at each point:

(i) Is $\frac{\partial f}{\partial x}$ positive, negative, or 0?
(ii) Which of $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ is bigger, and how do we arrange the values of $\frac{\partial f}{\partial x}$ (and likewise for $\frac{\partial f}{\partial y}$) in increasing order at all of these points?

For (i), the label of the contour is always increasing as we move from west to east (i.e., in the direction of increasing x), so $\frac{\partial f}{\partial x} > 0$ at all of these points. Alternatively, we can compute this partial derivative to be $\frac{\partial f}{\partial x} = y^2 + 1$, which is visibly always positive.

Turning to (ii), at $P = (0, 0)$ if we pass through going north (the direction of increasing y) then we stay on a fixed contour line, so $\frac{\partial f}{\partial y}(0, 0) = 0$. Since $\frac{\partial f}{\partial x} > 0$ at all points, we know that $\frac{\partial f}{\partial x}(0, 0) > \frac{\partial f}{\partial y}(0, 0)$. At the other points, it is difficult to tell from the contour plot whether the contour lines are closer or further apart when we travel through the point from west to east versus when we pass through the point from south to north. However, at S and T the nearby contour lines (which again are shown for f-value increments of 1) are more tightly packed in both the vertical and horizontal directions than at P, Q, R, so

$$f_x(S), f_x(T) > f_x(P), f_x(Q), f_x(R)$$

(and the symmetry of Figure 9.5.5 across the x-axis suggests that $f_x(S) = f_x(T)$).

For f_y, we similarly see that the steepness (whether up or down) in the vertical direction looks greater at S and T than at P, Q, R, so

$$|f_y(S)|, |f_y(T)| > |f_y(P)|, |f_y(Q)|, |f_y(R)|.$$

The reason for the absolute values is that we haven't yet taken into account the sign of the y-partial (whereas we have already seen that $f_x > 0$ everywhere). Looking at the contour labels, we see that f is decreasing as we pass through T moving upwards (i.e., labels of contour lines nearby are going down), so $f_y(T) < 0$, whereas f is increasing as we pass through S moving upwards, so $f_y(S) > 0$. Likewise, $f_y \approx 0$ near P, Q, R since the contour lines are rather "spread out" when moving vertically through those points. Hence, the picture provides the qualitative conclusion

$$f_y(S) > f_y(P), f_y(Q), f_y(R) > f_y(T).$$

Let's compare these qualitative conclusions from the contour plot with exact calculations of partial derivatives from the formula for f. We have $\partial f/\partial x = y^2 + 1$, so

$$\frac{\partial f}{\partial x}(0,0) = 1 \quad \frac{\partial f}{\partial x}\left(\frac{3}{2}, 0\right) = 1 \quad \frac{\partial f}{\partial x}\left(\frac{7}{2}, 0\right) = 1 \quad \frac{\partial f}{\partial x}(2,1) = 2 \quad \frac{\partial f}{\partial x}(2,-1) = 2.$$

Similarly $\partial f/\partial y = 2xy$, so

$$\frac{\partial f}{\partial y}(0,0) = 0 \quad \frac{\partial f}{\partial y}\left(\frac{3}{2}, 0\right) = 0 \quad \frac{\partial f}{\partial y}\left(\frac{7}{2}, 0\right) = 0 \quad \frac{\partial f}{\partial y}(2,1) = 4 \quad \frac{\partial f}{\partial y}(2,-1) = -4.$$

These exact calculations agree with the qualitative conclusions we obtained from inspection of the contour plot. ■

9.6. Second partial derivatives. There is a notion of "second derivative" for multivariable functions $f : \mathbf{R}^n \to \mathbf{R}$, defined similarly to the case of single-variable functions: we compute a partial derivative of a partial derivative. The new issue when $n > 1$ is that we have the freedom to *choose* in which direction to form each successive partial derivative.

As an example with $n = 2$, consider

$$f(x,y) = x^3 - 7x^2y + 5y^4.$$

Note that $f_x = \partial f/\partial x$ is equal to $3x^2 - 14xy$ and $f_y = \partial f/\partial y$ is equal to $-7x^2 + 20y^3$. Now we have the following options for a second derivative: we can differentiate f_x with respect to x or with respect to y, or we can differentiate f_y with respect to x or with respect to y.

We introduce notation for all of these possibilities:

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right) \text{ denotes the } x\text{-partial derivative of } f_x;$$

$$\frac{\partial^2 f}{\partial y \partial x} = \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial x}\right) \text{ denotes the } y\text{-partial derivative of } f_x;$$

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial y}\right) \text{ denotes the } x\text{-partial derivative of } f_y;$$

$$\frac{\partial^2 f}{\partial y^2} = \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial y}\right) \text{ denotes the } y\text{-partial derivative of } f_y;$$

Example 9.6.1. For $f(x,y) = x^3 - 7x^2y + 5y^4$ as above, we have computed f_x and f_y, so we can differentiate each of those with respect to either x or y to compute all 4 second derivatives:

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x}(3x^2 - 14xy) = 6x - 14y, \quad \frac{\partial^2 f}{\partial y \partial x} = \frac{\partial}{\partial y}(3x^2 - 14xy) = -14x,$$

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial}{\partial x}(-7x^2 + 20y^3) = -14x, \quad \frac{\partial^2 f}{\partial y^2} = \frac{\partial}{\partial y}(-7x^2 + 20y^3) = 60y^2.$$

■

Example 9.6.2. Let $f(x,y) = x\ln(1+y)$. Then $f_x = \ln(1+y)$ whereas $f_y = x/(1+y)$. Therefore,

$$\frac{\partial^2 f}{\partial x^2} = 0, \quad \frac{\partial^2 f}{\partial y \partial x} = \frac{\partial}{\partial y}(\ln(1+y)) = \frac{1}{1+y},$$

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial}{\partial x}\left(\frac{x}{1+y}\right) = \frac{1}{1+y}, \quad \frac{\partial^2 f}{\partial y^2} = \frac{-x}{(1+y)^2}.$$

■

You will notice that in both of the preceding examples, we have

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x},$$

or in words: the derivative of f_y with respect to x is the same as the derivative of f_x with respect to y. This possibly surprising result holds in all reasonable situations. It is called *equality of mixed partials*, and implies that for a function $f(x, y)$ there are really only 3 second partial derivatives (namely $\frac{\partial^2 f}{\partial x^2}$, $\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x}$, and $\frac{\partial^2 f}{\partial y^2}$) and not 4 second partial derivatives as initially seemed to be the case.

These concepts make sense for functions $f(x_1, \dots, x_n)$ of n variables for any n (not just $n = 2$):

Definition 9.6.3. For a function $f(x_1, \dots, x_n)$ of n variables that is differentiable in each x_i separately (as happens in all examples that arise for us), the *second partial derivatives* are defined to be

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = x_i\text{-partial derivative of } \frac{\partial f}{\partial x_j} \quad \left(\text{when } j = i \text{ it is denoted } \frac{\partial^2 f}{\partial x_i^2}\right)$$

when these exist.

In accordance with the "equality of mixed partials" that we observed in Examples 9.6.1 and 9.6.2, we have as a general (perhaps surprising) fact:

Theorem 9.6.4 (Clairaut–Schwarz). Consider a function $f(x_1, \dots, x_n)$ that is continuous, and for $1 \le i, j \le n$ suppose that the partial derivatives $\partial f/\partial x_i$ and $\partial f/\partial x_j$ as well as the second partial derivatives $\partial^2 f/\partial x_i \partial x_j$ and $\partial^2 f/\partial x_j \partial x_i$ exist and are continuous (as will be the case in all examples that arise for us). Then the order of applying $\partial/\partial x_i$ and $\partial/\partial x_j$ to f does not matter:

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i}.$$

(Due to the equality, we often denote this second partial derivative by the notation $f_{x_i x_j}$ or $f_{x_j x_i}$.)

For example, in the case of a 3-variable function $f(x, y, z)$, equality of mixed partials says that there are really only 6 possibly "different" second partial derivatives:

$$\frac{\partial^2 f}{\partial x^2}, \frac{\partial^2 f}{\partial y^2}, \frac{\partial^2 f}{\partial z^2}, \frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x}, \frac{\partial^2 f}{\partial x \partial z} = \frac{\partial^2 f}{\partial z \partial x}, \frac{\partial^2 f}{\partial y \partial z} = \frac{\partial^2 f}{\partial z \partial y}.$$

A precise explanation for Theorem 9.6.4 is provided in Section 9.7 for those who are interested. There we also give an example to illustrate that the equality can fail for somewhat peculiar $f(x, y)$ whose graph doesn't look at all strange near the point where the equality breaks down (see Figure 9.7.1).

Remark 9.6.5. Theorem 9.6.4 is true more for algebraic than geometric reasons, though its original discovery (without proof) by N. Bernoulli[12] around 1718 in a geometric setting ([En, App. 1, p. 202, last 3 lines], esp. "$\text{đ}\delta = \delta\text{đ}$" there) led him to consider it to be "an axiom, which I thought to be obvious to anybody" [En, Sec. 4.2.5]. Clairaut's proof in 1740 was wrong, as were many subsequent ones. Only in the 1860's was it realized that the result requires some hypotheses on the mixed partials. The first correct proof by modern standards was given by Schwarz in 1873 [**En**, Sec. 1.4, pp. 11-12]. (The history is complicated due to tremendous changes in the practice of mathematics over the past 300 years.)

---

[12] Nicolaus I Bernoulli (1687-1759) belonged to a family of many famous 17th and 18th century Swiss mathematicians and made important contributions to the early development of probability theory and differential equations. He rarely published his results, preferring to communicate them via private letters, because (as he wrote in a letter [En, Sec. 4.1.1, p. 95]) "I am not ambitious, and [...] have judged it to be sufficient that [...] I have solved it."

9.7. Proof of equality of mixed partials. To explain more why Theorem 9.6.4 holds, we focus on the case $n = 2$ to simplify the notation; this is the essential case anyway, since the result concerns partial derivatives in two coordinate directions (the other coordinates are fixed throughout the process). So consider a continuous $f(x, y)$ for which $\partial f/\partial x$, $\partial f/\partial y$, $\partial^2 f/\partial x\partial y$, $\partial^2 f/\partial y\partial x$ all exist and are continuous. Our goal is to show that

$$\frac{\partial^2 f}{\partial x\partial y} = \frac{\partial^2 f}{\partial y\partial x}.$$

First we give an informal reason for equality of mixed partials goes, again in the case $n = 2$ (which is really the essential case). For h, k near zero, the limit definition of partial derivatives gives

$$\frac{\partial^2 f}{\partial y\partial x}(a,b) = \frac{\partial(f_x)}{\partial y}(a,b) \approx \frac{f_x(a,b+k) - f_x(a,b)}{k},$$

$$f_x(a,b+k) \approx \frac{f(a+h,b+k) - f(a,b+k)}{h}, \quad f_x(a,b) \approx \frac{f(a+h,b) - f(a,b)}{h}.$$

Feeding the second and third approximations into the first yields

$$\begin{aligned}\frac{\partial^2 f}{\partial y\partial x}(a,b) &\approx \frac{(f(a+h,b+k) - f(a,b+k))/h - (f(a+h,b) - f(a,b))/h}{k} \\ &= \frac{f(a+h,b+k) - (f(a,b+k) + f(a+h,b)) + f(a,b)}{hk},\end{aligned}$$

and the same considerations with the roles of $\partial/\partial x$ and $\partial/\partial y$ swapped yields

$$\frac{\partial^2 f}{\partial x\partial y}(a,b) \approx \frac{f(a+h,b+k) - (f(a+h,b) + f(a,b+k)) + f(a,b)}{hk}.$$

The expressions approximating the two mixed partials are equal since $f(a,b+k) + f(a+h,b) = f(a+h,b) + f(a,b+k)$, so taking h, k ever closer to 0 then yields the equality of the limits of these approximations, namely equality of $(\partial^2 f/\partial y\partial x)(a,b)$ and $(\partial^2 f/\partial x\partial y)(a,b)$ as desired.

This argument, due to Euler around 1730 [**En**, Sec. 1.4, pp. 10-11, App. 2], was the first one for Theorem 9.6.4. It can be turned into an actual proof by handling the notation "≈" with more precision (to make contact with the continuity hypotheses that cannot be dropped).

A more complete justification, based on other principles (closer to the level of this course), goes as follow. Upon *fixing* the value of x, we first apply the Fundamental Theorem of Calculus to $g(y) = f(x, y)$. Note that g is differentiable with derivative $g'(y) = (\partial f/\partial y)(x, y)$, by the very definition of y-partial derivatives; keep in mind that we have *fixed the value of* x. By hypothesis the partial derivative f_y is continuous, so g' is continuous. Hence, the Fundamental Theorem of Calculus applies to g, yielding that for any y_0 we have

$$f(x, y_0) = g(y_0) = g(0) + \int_0^{y_0} g'(y)\,dy = f(x, 0) + \int_0^{y_0} \frac{\partial f}{\partial y}(x, y)\,dy. \tag{9.7.1}$$

Now consider differentiating both sides of (9.7.1) with respect to x. Approximate the definite integral with respect to y by a Riemann sum. If we x-differentiate each term in the Riemann sum and then pass to the limit of those Riemann sums, since x-differentiation passes through sums it is plausible that we get the x-partial of the limit of the Riemann sums, which is the x-partial of the definite integral with respect to y. This says $\partial/\partial x$ "passes through" the definite integral with respect to y, or in words: we can differentiate with respect to x and integrate with respect to y *in either order*. (A rigorous justification uses continuity of $\frac{\partial}{\partial x}(\partial f/\partial y) = \partial^2 f/\partial x \partial y$, so this is where the continuity of $\partial^2 f/\partial x\partial y$ is used in the reasoning.) Applying this fact, taking the partial derivative of both sides of (9.7.1) with respect to x yields

$$\frac{\partial f}{\partial x}(x, y_0) = \frac{\partial f}{\partial x}(x, 0) + \int_0^{y_0} \frac{\partial^2 f}{\partial x \partial y}(x, y)\, dy.$$

This last equality is valid for any x and any value of y_0. To go further we want to vary the value of y_0 and so treat it as a variable, which we wish to denote as y. Thus, to avoid conflict of notation with the integration variable on the right side, we rename the latter as t. That is, we restate the formula we have as:

$$\frac{\partial f}{\partial x}(x, y) = \frac{\partial f}{\partial x}(x, 0) + \int_0^{y} \frac{\partial^2 f}{\partial x \partial y}(x, t)\, dt. \tag{9.7.2}$$

Now apply $\partial/\partial y$ to both sides of (9.7.2). The left side becomes $\partial^2 f/\partial y \partial x$. On the right side, the y-derivative of the first term vanishes because this term does not depend on y, and we can use the Fundamental Theorem of Calculus in the y-direction for the second term to see that its y-derivative equals the integrand at $t = y$. (This application of the Fundamental Theorem of Calculus uses continuity of the integrand along the direction of integration, which in this case is the second coordinate direction; that in turn holds because of our continuity assumption on $\partial^2 f/\partial x \partial y$.) But that integrand is simply the second-partial $\partial^2 f/\partial x \partial y$, so equating left and right sides after applying $\partial/\partial y$ to each side yields $(\partial^2 f/\partial y \partial x)(x, y) = (\partial^2 f/\partial x \partial y)(x, y)$ as desired. We are done!

Just so you can see that the preceding result is not a formality, we give an explicit $f(x, y)$ in Example 9.7.1 below for which equality of mixed partials fails at the origin. This $f(x, y)$ violates the hypothesis in Theorem 9.6.4 that its second-partials are all continuous, yet its graph $z = f(x, y)$ shown in Figure 9.7.1 looks nice near the point where equality of mixed partials fails, so one can't expect to "see" such bad behavior just by staring at the graph.

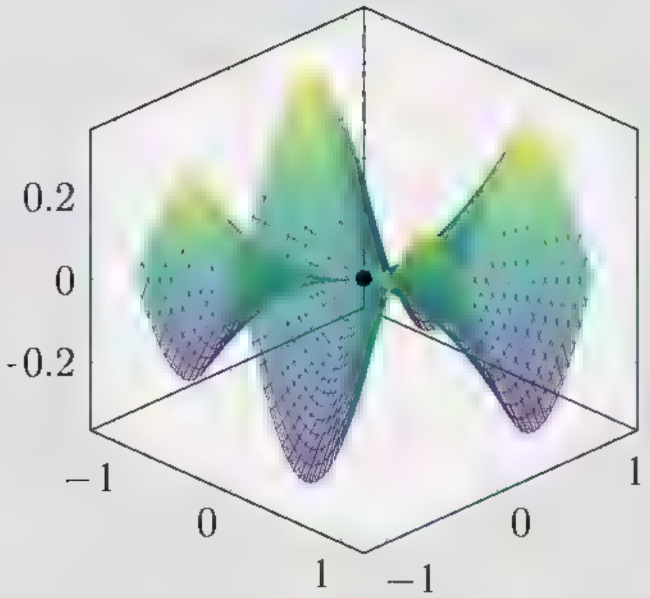


FIGURE 9.7.1. The graph of $f(x, y)$ violating equality of mixed partials

Example 9.7.1. Consider

$$f(x, y) = \begin{cases} xy(y^2 - x^2)/(x^2 + y^2), & (x, y) \neq (0, 0), \\ 0, & (x, y) = (0, 0). \end{cases}$$

Note that the roles of x and y here are slightly asymmetric because of the sign in the numerator. The graph of $f(x,y)$ looks a bit wavy but not otherwise so bad at the origin. It turns out that $(\partial^2 f/\partial x\partial y)(0,0) = -1$ and $(\partial^2 f/\partial y\partial x)(0,0) = 1$. To see this, we will compute $(\partial^2 f/\partial x\partial y)(0,0)$ by first calculating $f_y(x,0)$ for all x, then differentiate it with respect to x, and finally set $x = 0$.

When $x \neq 0$, we can use the formula defining f to compute $f_y(x,y)$ for $x \neq 0$ and any y. Doing this, and simplifying the resulting expression using standard algebra, one finds that $f_y(x,0) = -x$ for all $x \neq 0$. Meanwhile, for $x = 0$ we have

$$f_y(0,0) = \lim_{h \to 0} \frac{f(0,h) - f(0,0)}{h} = \lim_{h \to 0} \frac{0 - 0}{h} = 0,$$

so $f_y(0,0) = 0$. Hence, the formula $f_y(x,0) = -x$ actually holds for all x (allowing $x = 0$). A similar argument shows that $f_x(0,y) = y$ for all y.

Therefore,

$$\frac{\partial^2 f}{\partial x\partial y}(x,0) = \frac{\partial}{\partial x}(f_y(x,0)) = \frac{\partial}{\partial x}(-x) = -1$$

for all x, so $\frac{\partial^2 f}{\partial x\partial y}(0,0) = -1$. On the other hand, calculating the same way gives that $\frac{\partial^2 f}{\partial y\partial x}(0,y) = \frac{\partial}{\partial y}(y) = 1$ for all y, so $\frac{\partial^2 f}{\partial y\partial x}(0,0) = 1$. ■

Chapter 9 highlights (links to highlights in [illegible] and next chapters)

Notation	Meaning	Location in text
$f_{x_i}, \frac{\partial f}{\partial x_i}, \frac{\partial f}{\partial x_i}\Big\|_{(a_1,\dots,a_n)}$	partial derivative of $f(x_1,\dots,x_n)$ with respect to the variable x_i	Definition 9.2.2 ($n = 2$), box at end of Section 9.3
$\frac{\partial^2 f}{\partial x_i \partial x_j}, \frac{\partial^2 f}{\partial x_i^2}$	second partial derivatives of $f : \mathbf{R}^n \to \mathbf{R}$	Definition 9.6.3
$f_{x_i x_j}$	shorthand notation for second partial derivatives of $f : \mathbf{R}^n \to \mathbf{R}$	end of Theorem 9.6.4

Concept	Meaning	Location in text
single-variable derivative in a linear approximation	for $f : \mathbf{R} \to \mathbf{R}$, $f(c+h) \approx f(c) + f'(c)h$ for small h and $f(x) \approx f(c) + f'(c)(x-c)$ for x near c	end of item (i) after (9.1.1)
partial derivative	for $f : \mathbf{R}^n \to \mathbf{R}$, hold all but one variable constant and differentiate in the remaining variable	(9.2.1), (9.2.2) for $n = 2$
second partial derivative	partial derivative of partial derivative (can use different coordinate directions at each step)	Definition 9.6.3

Result	Meaning	Location in text
equality of symbolic and numerical methods for computing partial derivatives	identify partial derivative as single-variable derivative in a specific coordinate direction	box at end of Section 9.3
basic geometry of surface graph is encoded in contour plot	the approximate magnitude and the sign of partial derivatives of $f : \mathbf{R}^2 \to \mathbf{R}$ are encoded in density of level curves and change of their labels when moving up or to the right across a contour plot with uniform increments for f-values	box preceding Example 9.5.1
equality of mixed partials	order of applying $\partial/\partial x_i$ and $\partial/\partial x_j$ doesn't matter	Theorem 9.6.4

Skill	Location in text
compute partial derivatives by both the symbolic and numerical methods	Examples 9.3.1, 9.3.2
in a contour plot for $f : \mathbf{R}^2 \to \mathbf{R}$ (with uniform increments for labels), use density and sparsity of level curves when moving horizontally or vertically through (a,b) to see if partial derivative at (a,b) has big or small absolute value (i.e., is f changing rapidly or not in that direction through (a,b)?)	Section 9.5
use increase or decrease in labels of level curves when moving horizontally or vertically through (a,b) in contour plot for $f : \mathbf{R}^2 \to \mathbf{R}$ to read off sign of partial derivative there (i.e., is f increasing or decreasing in that direction through (a,b)?)	Section 9.5
compute second partial derivatives of $f : \mathbf{R}^n \to \mathbf{R}$	Examples 9.6.1, 9.6.2

10. Maxima, minima, and critical points

A useful result in single-variable calculus is the *first derivative test*:

> if $f : (a, b) \to \mathbf{R}$ is differentiable and attains a local maximum or local minimum at some point $x = c$ inside this open interval (i.e., $a < c < b$) then $f'(c) = 0$. (Any point where $f'(c) = 0$ is called a *critical point* for f, so local maxima and local minima of differentiable functions on an open interval are always critical points.)

It is worth emphasizing that the differentiability of f is necessary here. For example, the function $f(x) = |x|$ attains its minimum value of 0 at $x = 0$, but since f is not differentiable at that point we cannot say that $x = 0$ is a critical point for f; the first derivative test is useless when dealing with functions that are not differentiable at all points.

In this chapter, we explore the situation for functions of more than one variable. Namely, we describe a first derivative test for (differentiable) *multivariable* functions, and then go on to define a notion of "critical point" for such functions. Things will be a bit more complicated than in the single variable case. Indeed, *optimization* (i.e., finding maxima and minima) of functions $f : \mathbf{R}^n \to \mathbf{R}$ contains a genuinely new phenomenon when $n \geq 2$: a new type of critical point intermediate between a local maximum and a local minimum, called a *saddle point*. As we will discuss more fully in Section 10.2, this is quite different from single-variable critical points at which the function attains neither a local maximum nor a local minimum (such as $f(x) = x^3$ at $x = 0$).

By the end of this chapter, you should be able to:

- use partial derivatives to identify *potential* local maxima and local minima (you **will not yet** be able to distinguish among these possibilities based on derivative information because a multivariable "second derivative test" is reached only much later, in Chapter 26);
- use a contour plot to distinguish local maxima, local minima, and saddle points.

10.1. Motivation. Optimization for functions of n variables is one of the important applications of multivariable calculus. We now describe some real-world situations where such optimization is used.

Example 10.1.1 (Function fitting, revisited). We discussed in Section 7.3 how linear algebra in $\mathbf{R}^N$ is used to find the line $y = mx + b$ that best fits N data points (x_i, y_i) (which we assume are not all on a common vertical line). More precisely, we say that this line "best fits" the data if the function

$$\mathrm{E}(m, b) = \sum_{i=1}^{N} (y_i - mx_i - b)^2$$

is minimized as a function of the two variables m, b.

Because $E(m, b)$ is a simple function (even though it looks messy, it is just a quadratic expression in m, b), there are various ways to determine where it reaches its minimum value. We shall do this using calculus in Example 10.3.1. This gives the same answer as if we had used only linear algebra (as it must), but the advantage of the calculus approach is that it can be used with more complicated fitting problems.

For example, suppose we look at the graph of the data points (x_i, y_i), and notice that it looks approximately "periodic" (like the graph of sine or cosine) as in Figure 10.1.1.

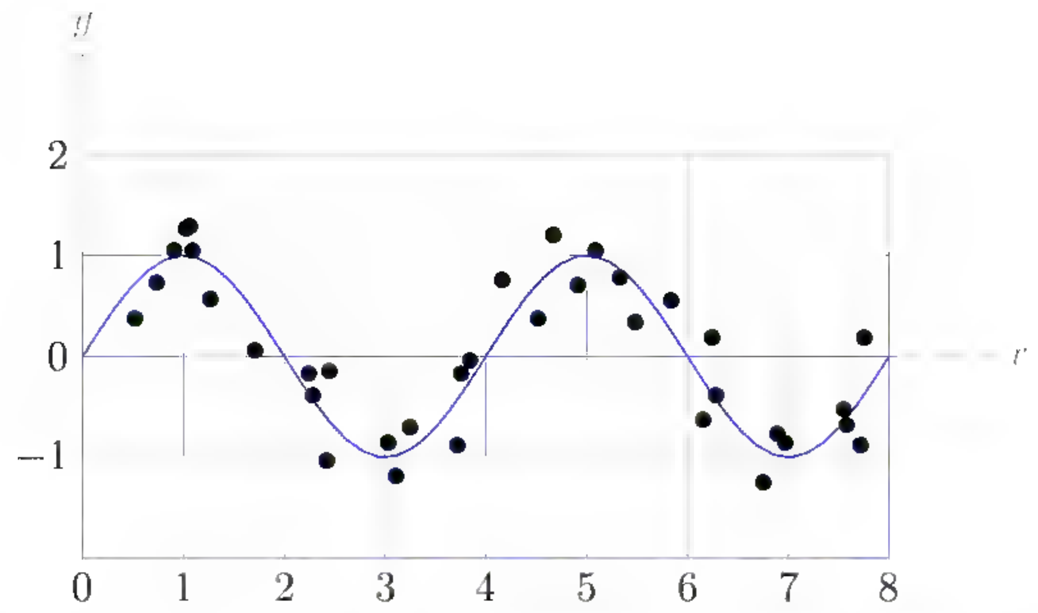


FIGURE 10.1.1. Raw data which appears to be approximately periodic

This might suggest that it is more reasonable to try to fit the data points (x_i, y_i) with a function of the form

$$A\sin(Bx + C),$$

The meaning of these variables is that B describes the frequency of the oscillation, A its amplitude, and C its phase.

Our fitting task amounts to finding (A, B, C) that minimizes the "sum of squared error" function

$$\mathrm{E}(A,B,C) = \sum_{i=1}^{N}(y_i - A\sin(Bx_i + C))^2.$$

This more complicated minimization problem can no longer be carried out using linear algebra, but requires multivariable calculus. ■

Example 10.1.2 (drug design, protein folding). In the physical sciences, many problems can be studied by means of energy minimization. For example, predicting how a protein folds is a fundamental problem in molecular biology, and a useful clue for understanding the folding that occurs in nature is that it minimizes the energy. The energy of a protein configuration corresponds to a function of hundreds or thousands of variables (that keep track of the configuration), and numerically solving the associated minimization problem is a valuable tool in work on protein folding. The geometric structure of a molecule also minimizes electronic energy, and the corresponding minimization problem is therefore of much interest in computational chemistry; see Appendix J.

In the field of computer-aided drug design, computer simulations based on energy minimization are used to predict which chemical compounds are likely to be worth creating in the lab to test out for a specific drug. ■

Example 10.1.3 (system of springs). Consider the system of hanging springs as shown in Figure 10.1.2. There are two springs, with spring constants k_1, k_2 holding up two weights with masses m_1, m_2. The two springs have length 1 when unstretched.

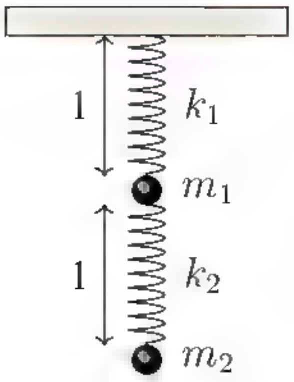


FIGURE 10.1.2. Mechanical configuration with two springs and two masses

Imagine that you initially support the weights with your hand, so none of the springs are stretched, and then you let go. What happens? The weights will drop down and then start bouncing up and down; eventually they will settle down to a new position (where the downward forces on the masses due to gravity are balanced by the upward forces on the masses due to the springs). To find this position, one can minimize a "potential energy" function given as follows. Write the lengths of the two springs as $1 + x_1, 1 + x_2$ (so that $x_1 = 0$ corresponds to the first spring not being stretched at all, and similarly for the other spring.) The potential energy is given by

$$V(x_1, x_2) = \frac{1}{2}(k_1 x_1^2 + k_2 x_2^2) - m_1 g x_1 - m_2 g(x_1 + x_2)$$

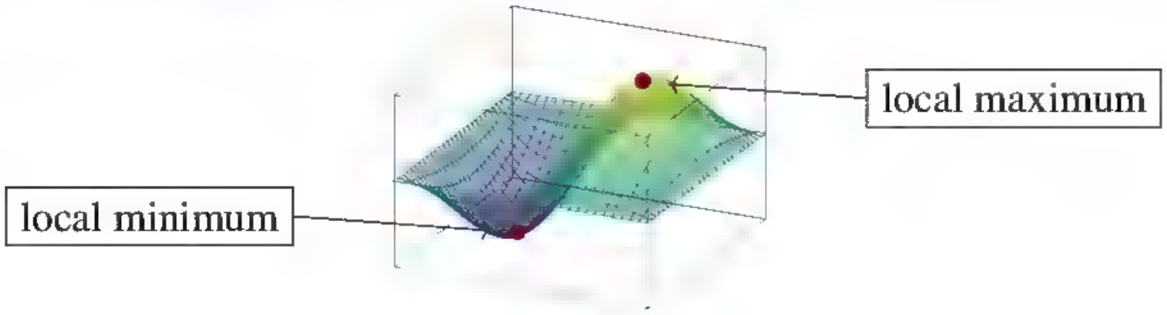
where g is a certain constant (the acceleration due to gravity at the surface of the earth). The sought-after position is the point (x_1, x_2) that minimizes $V(x_1, x_2)$. (This problem can be solved in other ways too.) A worked example of minimizing such a function is given in Example 10.2.14 (using a multivariable analogue of the "first derivative test" for extrema from single-variable calculus, to be introduced in Section 10.2).

We can *also* analyze the bouncing. This is an example of a system of *coupled oscillators*, and we will show in Section 27.1 how to analyze such systems using some more advanced techniques in linear algebra to be developed later in this book. An interesting feature of such applications is that the physical process is governed by a system of *differential equations* yet linear algebra turns out to be an extremely powerful tool for organizing the solution process there too. ■

10.2. Testing for critical points. Just as for functions of one variable, we need to define what it means for a function $f : \mathbf{R}^n \to \mathbf{R}$ of n variables to have a local maximum or a local minimum. For simplicity, suppose that $n = 2$ (though the general definition is the same, only with a bit more notation).

Definition 10.2.1. A function $f(x, y)$ achieves a *local maximum* at (a, b) if $f(a, b) \geq f(x, y)$ for all (x, y) which are sufficiently near to (a, b). In other words, if we move in any direction from (a, b), then as long as we stay nearby, $f(x, y)$ *decreases* (or stays the same).

A function $f(x, y)$ achieves a *local minimum* at (a, b) if $f(a, b) \leq f(x, y)$ for all (x, y) sufficiently near (a, b) (i.e., moving in any direction from (a, b) causes $f(x, y)$ to *increase* or stay the same).



Let us try to understand what is going on by reducing to functions of one variable. Suppose that $f(x, y)$ has a local maximum at (a, b), in the sense of Definition 10.2.1. If we keep x close to a and keep y fixed at the value b then (as we discussed in Chapter 9) we have

$$f(a+h, b) \approx f(a, b) + f_x(a, b)h$$

for values of h near 0.

Thus, if $f_x(a, b) > 0$ and we consider small $h > 0$ then this shows that $f(a + h, b)$ should be larger than $f(a, b)$, so (a, b) cannot be a local maximum. Similarly, if $f_x(a, b) < 0$ and we consider small $h < 0$ then $f(a + h, b)$ will again be larger than $f(a, b)$. (This is analogous to the fact that if the derivative for a single-variable function is positive somewhere then the graph is sloping upward nearby, while if the derivative is negative somewhere there then the graph is sloping downward nearby.)

This shows that if (a, b) is a local maximum then the options $f_x(a, b) > 0$ and $f_x(a, b) < 0$ are *ruled out*, so necessarily $f_x(a, b) = 0$. Using precisely the same reasoning by wiggling y while keeping $x = a$, when (a, b) is a local maximum for f we also conclude also that $f_y(a, b) = 0$.

We summarize what we have learned as follows:

$$\text{if } f : \mathbf{R}^2 \to \mathbf{R} \text{ has a local maximum at } (a, b)\text{, then } \frac{\partial f}{\partial x}(a, b) = 0 \text{ and } \frac{\partial f}{\partial y}(a, b) = 0.$$

We can reason the same way for local minima, and also for functions of more than two variables. This leads to a general result:

Let $f : \mathbf{R}^n \to \mathbf{R}$ be a function.

Theorem 10.2.2. Suppose that a point $\mathbf{a} \in \mathbf{R}^n$ is either a local maximum or a local minimum of f. Then all partial derivatives of f vanish at $\mathbf{x} = \mathbf{a}$; i.e., $\dfrac{\partial f}{\partial x_i}(\mathbf{a}) = 0$ for $1 \leq i \leq n$.

Definition 10.2.3. If $\dfrac{\partial f}{\partial x_i}(\mathbf{a}) = 0$ for all $1 \leq i \leq n$ then we say $\mathbf{a}$ is a *critical point* for f. In particular, every local maximum and every local minimum of $f : \mathbf{R}^n \to \mathbf{R}$ is a critical point. (In Chapter 26 you will learn a "multivariable second derivative test" that gives a way to check if a critical point is a local extremum. You are *not* expected to carry out this important further task until much later in the course.)

Remark 10.2.4. Strictly speaking, in the preceding theorem and definition we should assume f is "differentiable" in an appropriate n-variable sense that recovers the notion from single-variable calculus when $n = 1$. We have not given that definition because it requires certain concepts in linear algebra that have not yet been introduced. However, it turns out that this definition (which we will give in Remark 13.5.13 for those who are interested) is always satisfied when the partial derivatives $\partial f/\partial x_i : \mathbf{R}^n \to \mathbf{R}$ all exist and are continuous, as will be the case for everything we do in this course.

Let us illustrate Theorem 10.2.2 and Definition 10.2.3 with a few examples.

Example 10.2.5. Consider $f(x, y) = x^2 + y^2$. We compute that

$$f_x(x, y) = 2x, \; f_y(x, y) = 2y,$$

so the simultaneous vanishing of partial derivatives happens only at the origin $(x, y) = (0, 0)$. Now $f(0, 0) = 0$, and if (x, y) is any other point (it does not even need to be near the origin), then $f(x, y) > 0$ by inspection. This means that $(0, 0)$ is a local minimum. Since the condition $f(x, y) \geq f(0, 0)$ even holds without requiring (x, y) to be close to $(0, 0)$, we say that $(0, 0)$ is a "global minimum" for f.

Similarly, if we consider $g(x, y) = -x^2 - y^2$ then the partial derivatives are $g_x = -2x$, $g_y = -2y$ and these simultaneously vanish only when $(x, y) = (0, 0)$. Furthermore, $g(0, 0) = 0$ while $g(x, y) < 0$ for any other point, so the origin is a global maximum for g.

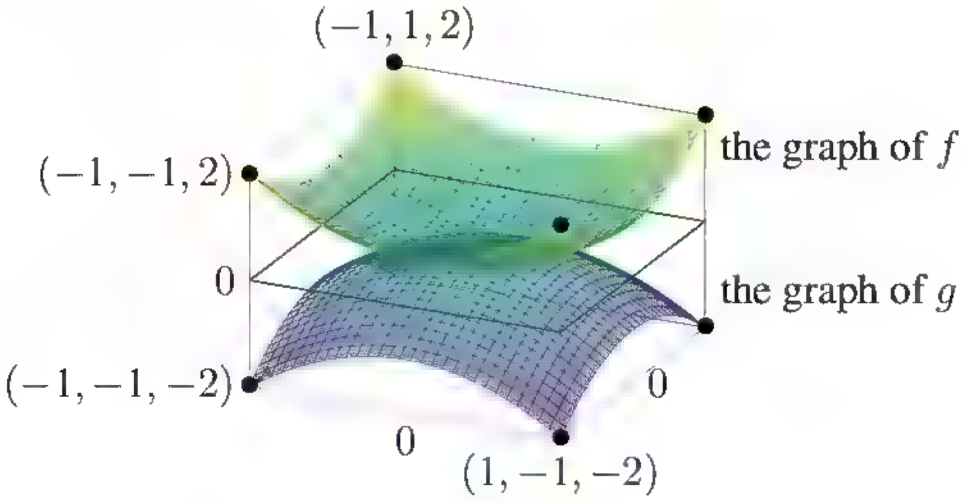


FIGURE 10.2.1. The graphs of $f = x^2 + y^2$ and $g = -x^2 - y^2$ over the square $[-1, 1] \times [-1, 1]$

These generalize to functions of more than 2 variables, as follows. The function

$$f(x_1, \ldots, x_n) = x_1^2 + \ldots + x_n^2$$

satisfies $f_{x_i} = 2x_i$ for each $i = 1, \ldots, n$. The partial derivatives simultaneously vanish only at the origin $(0, \ldots, 0)$, and just as before, the origin is a global minimum (since a sum of squares must be ≥ 0 and can only vanish when the squares all equal 0). Likewise the function $g(x_1, \ldots, x_n) = -x_1^2 - \ldots - x_n^2$ has a global maximum at the origin.

The extrema for these functions can be understood without (partial) derivatives, by the algebraic reasons mentioned above. Also, for $n = 2$ the graph can be visualized as "bowl shaped", pointing either down or up (depending on the sign), making it visually apparent that the origin is a local maximum or local minimum (depending on the sign). In subsequent examples we will have to do more work to find local maxima and local minima. ■

Example 10.2.6. Let $g(x, y) = 3x^2y + 2y^3 - xy$, so

$$g_x = 6xy - y, \quad g_y = 3x^2 + 6y^2 - x,$$

so for a point (a, b) to be a critical point the conditions are $0 = 6ab - b = b(6a - 1)$ and $3a^2 + 6b^2 = a$. The first condition says that either $b = 0$ or $a = 1/6$, and correspondingly the second condition says $3a^2 = a$ (so $a = 0$ or $a = 1/3$) or $3(1/6)^2 + 6b^2 = 1/6$ respectively. In the latter case, we have $6b^2 = 1/12$, so $b = \pm 1/(6\sqrt{2})$. Putting it all together, there are 4 critical points:

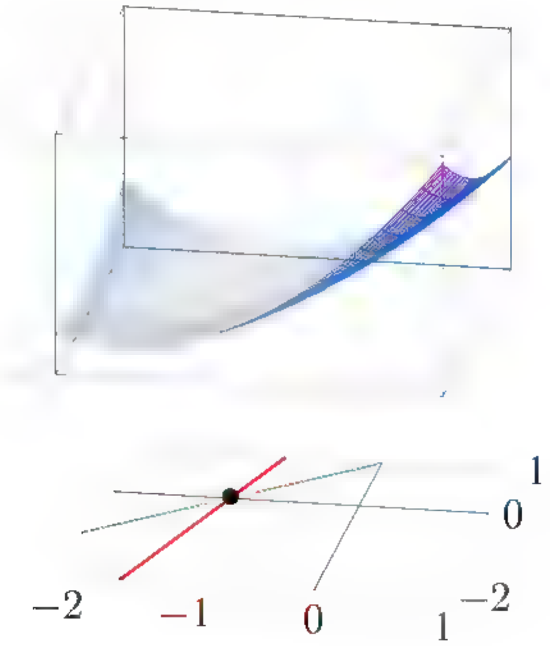
$$P = (0, 0), \quad Q = (1/3, 0), \quad R = (1/6, 1/(6\sqrt{2})), \quad S = (1/6, -1/(6\sqrt{2})).$$

Are these local extrema? It turns out that R is a local minimum, S is a local maximum, and P and Q are of neither type! We will return to this to analyze it via a *multivariable second derivative test* based on linear algebra in Section 26.4 (see Example 26.2.3, with Figure 26.2.2 showing a contour plot for g). ■

Example 10.2.7. Let $f(x, y) = 4x^2 - 2xy + y^2 + 8x - 2y + 5$. We compute that

$$f_x = 8x - 2y + 8, \; f_y = -2x + 2y - 2.$$

Solving $f_x = f_y = 0$ amounts to solving a system of 2 linear equations in 2 unknowns. Using substitution of one equation into the other, the unique solution is $(x, y) = (-1, 0)$. This is the only critical point of f, at which the value is $f(-1,0)=1$.



The surface is the graph of

$$f(x,y)=4x^2-2xy+y^2+8x-2y+5.$$

The green line is determined by the equation

$$f_y=-2x+2y-2=0.$$

The red line is determined by the equation

$$f_x=8x-2y+8=0.$$

Their point of intersection is $(x,y)=(-1,0)$.

How can we tell whether this is a local maximum or local minimum, or perhaps neither? That is, how do the values of $f(x,y)$ for (x,y) near $(-1,0)$ compare to the value 1 at $(-1,0)$? One method is: plot the graph on a computer (as shown above) and just stare at it! By visual inspection, it looks like it is probably a local minimum, though to be more convinced one might want to "zoom in" and rotate the graph a bit to be sure. This method has a couple of defects, the most serious one being that it doesn't adapt well to problems in more unknowns (we can't literally "see" the graph of $f(x_1,\ldots,x_n)$ in $\mathbf{R}^{n+1}$ when $n>2$). We want to use methods that can adapt to problems with any number of variables, so we seek another technique.

In Chapter 26 we will discuss a very systematic method applicable in any number of variables – essentially a multivariable version of the second derivative test – but for now we handle it using a bit of algebra (which adapts to quadratic expressions in any number of variables; you are not expected to figure it out for yourself yet, but you can read through the calculations when presented as below).

The idea is first to move the coordinates to be centered around the point of interest $(-1,0)$. Define $v=x+1$ and $w=y$, so in (v,w)-coordinates that point becomes the origin. The usefulness of this is seen by rewriting everything in terms of v and w rather than x and y: since $x=v-1$ and $y=w$, we can write

$$\begin{aligned}f(x,y)=f(v-1,w)&=4(v-1)^2-2(v-1)w+w^2+8(v-1)-2w+5\\&=4v^2-8v+4-2(vw-w)+w^2+8v-8-2w+5\\&=4v^2+w^2-2vw+1.\end{aligned}$$

This eliminated the appearance of "degree 1" terms (there is no $av+bw$ term).

Now comes the part which we do not expect you to see on your own, which is to *complete the square* in a clever way: by splitting up $4v^2$ into v^2 (to be absorbed with w^2-2vw to make $w^2-2vw+v^2=(w-v)^2$ that is a perfect square) and a remaining $3v^2$ we get

$$4v^2-2vw+w^2+1=3v^2+(v^2-2vw+w^2)+1=3v^2+(v-w)^2+1$$

that is ≥ 1 everywhere, and it is only ever equal to 1 when the two terms $3v^2$ and $(v-w)^2$ that are always ≥ 0 actually vanish. Such vanishing forces $v=0$ and $v-w=0$, so also $w=0$, and hence only happens at the origin $(v,w)=(0,0)$. That corresponds to being at $(x,y)=(-1,0)$, so we have shown by bare hands that $f(x,y)$ has a local (even global) *minimum* at $(-1,0)$. In Example 10.2.14 we'll give one more example resting on algebraic cleverness and luck, and in Chapter 26 we will develop more robust tools involving no cleverness and no luck. ■

Example 10.2.8. In Figure 10.2.2 we give a contour plot of $f(x, y) = x^2 - y^2$:

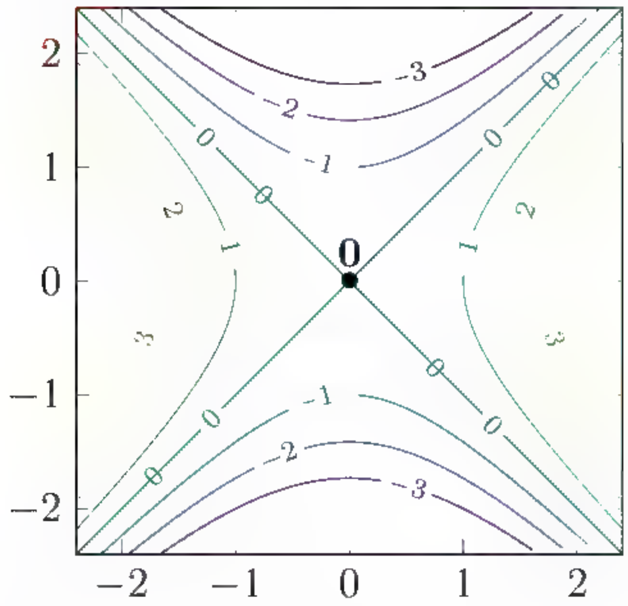


FIGURE 10.2.2. A contour plot for $f(x, y) = x^2 - y^2$ using increments of 1 in f-values

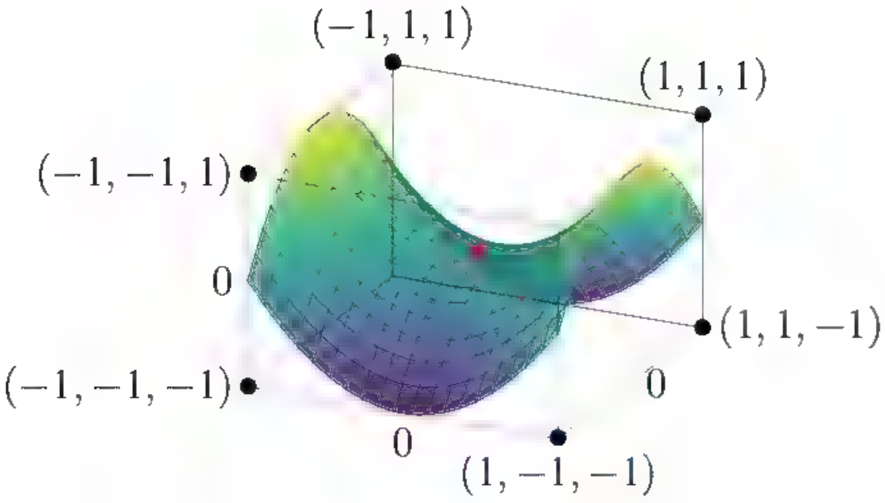
The level curves are labeled in f-value increments of 1. As we move along lines in $\mathbf{R}^2$ through the origin, is the function $f(x, y)$ increasing or decreasing, and does it have a local maximum or local minimum at the origin on the line of motion? This question is internal to the xy-plane $\mathbf{R}^2$, though it can also be expressed in terms of the surface graph $z = f(x, y)$ in $\mathbf{R}^3$: if we stand over the origin and walk in some straight-line direction (such as due north, or due east), are we going uphill or downhill?

Let's analyze this for walking due north and then due east, with three different ways of analysis: first using the contour plot, then using a computation of the partial derivatives of $f(x, y)$, and finally looking at the graph in $\mathbf{R}^3$. All three approaches will give consistent conclusions, but some ways will provide more refined information than others.

In the contour plot, the level set going through the origin is labeled by 0, and when heading north from the origin the contour lines (i.e., level sets) are labeled with decreasing negative numbers. This says that we are going downhill as we head north. Similarly, when heading east away from the origin we see that the contour curves are labeled with increasing positive numbers, so such a path is going uphill.

To compare those qualitative conclusions with exact calculations using partial derivatives, we first note that $\dfrac{\partial f}{\partial y} = -2y$. At the origin this value is zero, but as we head north (i.e., letting y increase beyond 0) this partial derivative becomes ever more negative. This means that a northerly path out of the origin begins approximately level but as we keep moving the terrain quickly goes downhill. Similarly, since $\dfrac{\partial f}{\partial x} = 2x$ we see that this vanishes at the origin but becomes ever more positive we as we head east (i.e., letting x increase beyond 0). Hence, an easterly path out of the origin begins approximately level but as we keep moving the terrain quickly goes uphill.

Finally, turning to the surface graph $z = x^2 - y^2$ in $\mathbf{R}^3$ as shown in Figure 10.2.3, we see that it has a "mountain pass" at the origin precisely because of the dichotomy in the behavior: $x^2 - y^2$ has a local minimum at the origin when restricted to certain lines through the origin in the xy-plane (such as the x-axis $y = 0$) and a local maximum at the origin when restricted to certain other lines through the origin in the xy-plane (such as the y-axis $x = 0$). The picture of the surface graph near $(0, 0, f(0, 0)) = (0, 0, 0)$ in Figure 10.2.3 makes clear why we call $(0, 0)$ a *saddle point* for $x^2 - y^2$.



The graph of $f(x,y) = x^2 - y^2$.

Notice the saddle point at $(x,y) = (0,0)$.

FIGURE 10.2.3. Graph $z = x^2 - y^2$ in $\mathbf{R}^3$, with red dot at a "saddle point" over $(0,0)$

Observe that in the contour plot for the same function $x^2 - y^2$ in Figure 10.2.2, at $(0,0)$ the plot has an "X" shape for the level set through that point. This will turn out to express exactly the "saddle" feature of the surface graph over $(0,0)$. We will soon define the concept of saddle point in general, and analyze it systematically (for any number of variables) throughout Chapter 26. It will be seen to represent a *genuinely new phenomenon* of the multivariable world with no counterpart in single-variable calculus, and has rather practical significance too (e.g., in Section J.2 it is related to a molecule transitioning between different states). ■

Next is an example involving functions that are slightly more complicated.

Example 10.2.9. Suppose that $h(x,y) = x\sin x - \cos y$. We compute that $h_x = \sin x + x\cos x$, $h_y = \sin y$, The equations $h_x = 0$ and $h_y = 0$ are now not so easy to solve explicitly. However, one solution is easy to find: namely $h_x(a,b)$ vanishes when $a = 0$, and $h_y(a,b)$ vanishes when $b = 0$, so $(0,0)$ is a critical point.

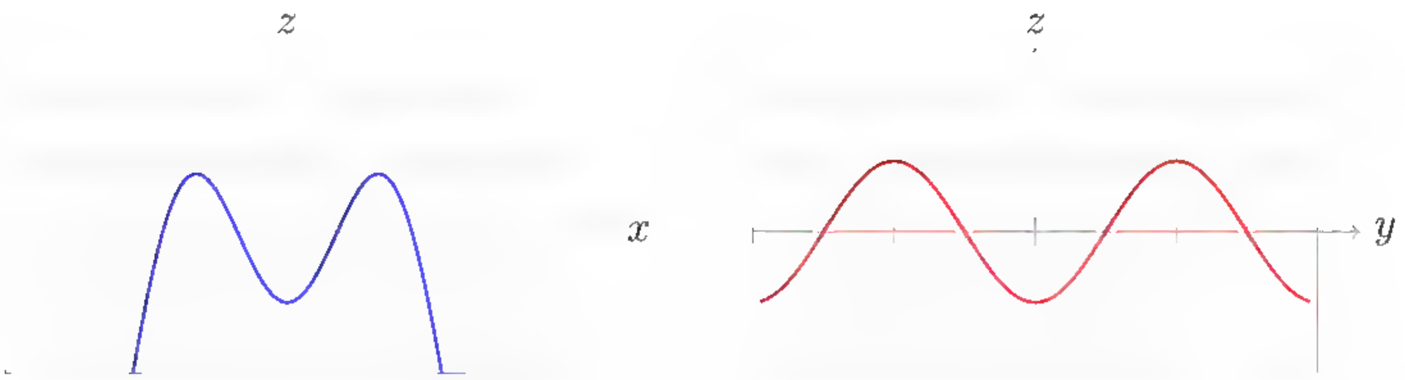


FIGURE 10.2.4. The graphs of $h(x,0) = x\sin x - 1$ and $h(0,y) = -\cos y$

By looking at the graph for $x\sin x - 1$ (see Figure 10.2.4) we see that this function has a local minimum as a function of x at $x = 0$, and $-\cos y$ has a local minimum as a function of y at $y = 0$. Thus, it seems reasonable to guess that h has a local minimum at $(0,0)$. This is true, and is suggested by the surface graph in Figure 10.2.5.

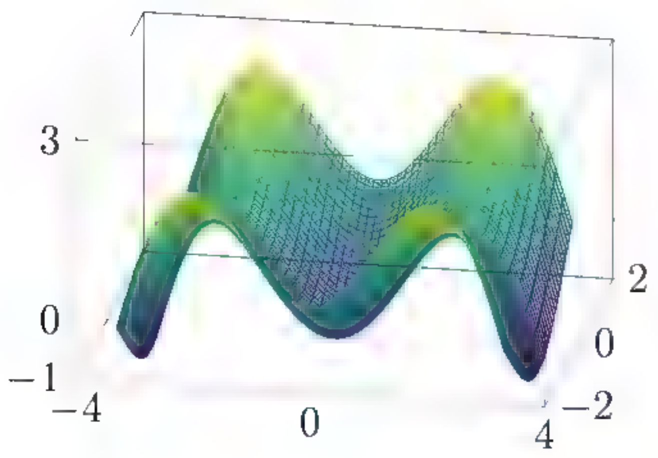


FIGURE 10.2.5. The graph of $h(x, y) = x \sin x - \cos y$

But looking at a surface graph is not a technique that helps beyond 2-variable functions. We will have tools of wider applicability to verify the local extremum property in Chapter 26 (see Example 26.2.1).

Note that h has many other critical points. The function $x \sin x - 1$ oscillates as $|x|$ grows (beyond the range shown in Figure 10.2.4), with successive peaks for large $|x|$ becoming bigger and bigger, so there are infinitely many values of x where its derivative vanishes. Similarly, $-\cos y$ has derivative $\sin y$ which vanishes at $y = 0, \pi, -\pi, 2\pi, -2\pi, \ldots$. So there are infinitely many critical points of h on $\mathbf{R}^2$. ■

We emphasize that *if* $\mathbf{a}$ *is a critical point of* $f : \mathbf{R}^n \to \mathbf{R}$*, then it may be neither a local maximum nor a local minimum.* In fact, for functions of two or more variables there is a new phenomenon (which we saw in Example 10.2.8 for $x^2 - y^2$ at $(0, 0)$):

Definition 10.2.10. A critical point $\mathbf{a} \in \mathbf{R}^n$ of $f : \mathbf{R}^n \to \mathbf{R}$ is a *saddle point* if (i) as we move away from $\mathbf{a}$ along some line (e.g., parallel to a coordinate axis) then f increases nearby, so $\mathbf{a}$ is a local minimum along that line, and (ii) as we move away from $\mathbf{a}$ along some other line then f decreases, so $\mathbf{a}$ is a local maximum along that line.

Such behavior can happen with n variables when $n > 1$ because then there are "more lines" in $\mathbf{R}^n$ through a point along which we can move away from the point: in $\mathbf{R}$ there is only one line through a point, but in $\mathbf{R}^2$ there are many lines through a point (e.g., not only horizontal, but also vertical, diagonal, etc.), let alone in $\mathbf{R}^n$ for $n > 2$. This is a *genuinely multivariable* phenomenon.

We do not yet have any general methods for determining when a critical point is a local maximum or local minimum or a saddle point (or perhaps even more exotic possibilities?). The method for this, which involve second partial derivatives, will be discussed in Chapter 26.

Example 10.2.11. For $f : \mathbf{R}^n \to \mathbf{R}$ with a critical point at $\mathbf{a} \in \mathbf{R}^n$, you may wonder why we have to wait until Chapter 26 to formulate a "multivariable second derivative test" when $n > 1$. Can't we somehow use the signs of the second partial derivatives of f at $\mathbf{a}$ (if these second partials at $\mathbf{a}$ are all nonzero) to figure out the behavior of f near $\mathbf{a}$? The answer is negative, as we shall illustrate in a moment with an example. The way around this difficulty requires organizing the values $f_{x_i x_j}(\mathbf{a})$ in terms of several new concepts in linear algebra (matrices, quadratic forms, eigenvalues) that are studied later in this book.

To illustrate why the signs of the second partials are insufficient when $n > 1$, we give a function $f(x, y)$ with a critical point $\mathbf{a}$ at which all three values $f_{xx}(\mathbf{a})$, $f_{yy}(\mathbf{a})$, $f_{xy}(\mathbf{a})$ are positive yet f has a saddle point (rather than a local minimum) at $\mathbf{a}$! Consider $f(x, y) = xe^{x+3y} - x + \sin^2(y)$. This has partials $f_x = (1 + x)e^{x+3y} - 1$ and $f_y = 3xe^{x+3y} + 2\sin(y)\cos(y) = 3xe^{x+3y} + \sin(2y)$ that both vanish at $\mathbf{0}$, so the origin is a critical point of f. We compute

$$f_{xx} = (2 + x)e^{x+3y}, \quad f_{yy} = 9xe^{x+3y} + 2\cos(2y), \quad f_{xy} = 3(1 + x)e^{x+3y},$$

so $f_{xx}(\mathbf{0}) = 2$, $f_{yy}(\mathbf{0}) = 2$, $f_{xy}(\mathbf{0}) = 3$ are all positive. On the coordinate axes we have $f(x, 0) = xe^x - x = x(e^x - 1)$ and $f(0, y) = \sin^2(y)$, each of which is checked (via single-variable calculus) to have a local minimum at the origin.

But the behavior of $f(x, y)$ on the line $y = -x$ is encoded in the function $g(x) = f(x, -x) = xe^{-2x} - x + \sin^2(-x) = x(e^{-2x} - 1) + \sin^2(x)$ that has a critical point at $x = 0$ (because $g'(x) = (1 - 2x)e^{-2x} - 1 + \sin(2x)$ satisfies $g'(0) = 0$) yet that is a local *maximum* for $g(x)$ since $g''(x) = 4(x - 1)e^{-2x} + \sin(4x)$ satisfies $g''(0) = -4 < 0$. Hence, f has a saddle point at $\mathbf{0}$. ■

Remark 10.2.12. Here is another genuinely multivariable phenomenon: any single-variable polynomial $a_nx^n + a_{n-1}x^{n-1} + \cdots + a_1x + a_0$ that is everywhere ≥ 0 attains a minimal value somewhere (e.g., think about a parabola, or $x^4 - 2x^2 + 3$), but this fails for multivariable polynomials! For example, $f(x, y) = (1 - xy)^2 + x^2 \geq 0$ and in fact $f(x, y) > 0$ everywhere (otherwise $f(a, b) = 0$ for some (a, b), but then $(1 - ab)^2$ and a^2 vanish, an impossibility since if $a = 0$ then $(1 - ab)^2 = 1^2 = 1 \neq 0$), yet f has *no minimal value* because for any $c > 0$ (however tiny you wish) we have $f(\sqrt{c}, 1/\sqrt{c}) = c$.

Example 10.2.13. The contour plot in Figure 10.2.6 for a function $f(x, y)$ shows several types of critical points. The numerical label on each contour line is the value of f along that contour line (and these appear with increments of 0.2 in the f-value). We want to find a local maximum, a local minimum, and a saddle point, and make some observations about the geometry of the contour plot near each type of point.

If we look at the collection of nested ovals centered on S, and inspect the numerical values of the function that label the level sets, we see that those numbers are strictly increasing, so ovals seem to be honing in on a point S that is a local maximum.

Likewise, the collection of nested ovals near P have function values going in a decreasing direction. Hence, we similarly expect that these ovals are honing in on a point P that is a local minimum.

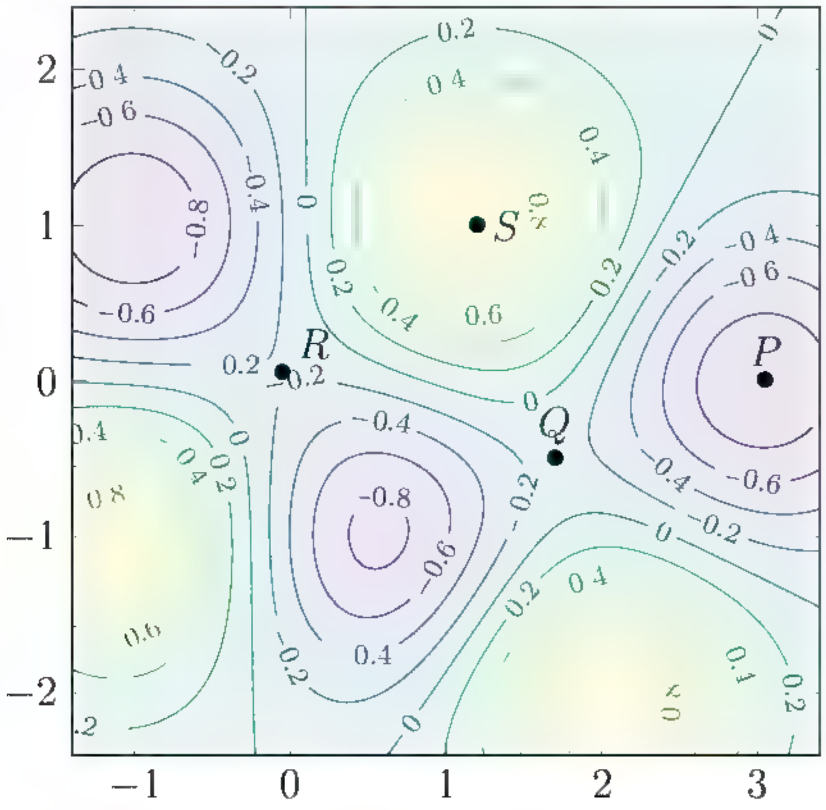


FIGURE 10.2.6. A contour plot with a variety of types of critical points

We also see saddle points at R and Q. Indeed, as we move away from R in the north-east direction then the value of the function (as indicated by the numerical label on the level sets nearby) is going *up* whereas if we move away from R in the south-east direction then the value of the function is going *down*.

There are (at least) two interesting observations to make:

(i) At a saddle point the contour plot looks like an "X" shape dividing the nearby space into 4 regions, with the saddle point at the center of the "X". The function values go up as we move away into one pair of opposite regions and go down as we move away into the other pair of opposite regions. On the other hand, near local extrema (both local maxima and local minima) the level sets are oval-shaped and in fact look increasingly like ellipses. Why is this?

(ii) Near the critical points, the locations of the level sets for a common small step size in the function value (.2 in the plot shown here) become rather "spread out"; i.e., not as densely packed as elsewhere. Why is this?

For (i), the first part of the observation is expressing approximately the geometry of the saddle shape of a surface graph as drawn in Example 10.2.8 for the function $x^2 - y^2$. The lesson is that $x^2 - y^2$ is a prototype for *all* 2-variable saddle points, with its contour plot (as shown in Example 10.2.8) a prototype for that near all saddle points of 2-variable functions. The reason this works so well as a prototype is ultimately due to a version of the multivariable second derivative test (Theorem 26.3.1), whose relevance will be explained in Section 26.3 (see Remark 26.3.5). This will also explain the observation about ellipses near local extrema, and an application of this mathematical understanding of saddle points to the transition process between molecular structures for a fixed set of atoms is given in Appendix J.

For (ii), one might try to reason in terms of the approximate flatness of a graph surface $z = f(x, y)$ as we get close to the top of a hill or the bottom of a valley (a 2-dimensional analogue of the flatness of the graph of a function $h(t)$ near a critical point in single-variable calculus). But there is another way to express the observation that will lead to an explanation having wider significance, as follows.

The visual picture in (ii) of spreading-out of level sets near a critical point is saying in numerical terms that at a place where the partial derivatives are all small, to attain a given numerical change in the function value towards a local extremum (such as increments of ± 0.15) seems to requires moving a *bigger* distance than at a place where the partial derivatives are not all small. This latter formulation also makes sense as a possible property for functions $f : \mathbf{R}^n \to \mathbf{R}$ near a critical point for *any* n (not just $n = 2$).

But is such a property for the behavior near a critical point actually true in general? Yes! You may try to convince yourself for $n = 2$ using hills and valleys in a surface graph; in Section 11.1 we will use the connection between dot products and angles among vectors (a synthesis of algebra and geometry) to give a general explanation for n-variable functions for any n (see Remark 11.1.6). ■

Example 10.2.14. Though we lack general tools at present (beyond the single-variable case) to determine whether or not a critical point is a local extremum, there are some "quadratic" cases where this can be done by hand. Let's work out a specific example: we will minimize the function

$$V(x, y) = 2x^2 + y^2 - 4x - 8(x + y);$$

a physical system where this arises is given in Example 10.1.3. To find the critical points, we compute

$$V_x = 4x - 12, \;\; V_y = 2y - 8,$$

so these simultaneously vanish only at $P = (3, 4)$. This is the unique critical point.

Is P a local maximum, local minimum, saddle point, or something worse? The physical context from Example 10.1.3 in terms of potential energy suggests that P should be a local minimum (since in physical settings the tendency is for systems to evolve in a direction that minimizes potential energy), but that isn't a justification. If we can give a mathematical justification then that would be a virtue of the mathematical model for the physical situation.

In lieu of any better idea, let's translate the coordinates to put the critical point P at the origin and see what happens. Define new variables $v = x - 3$ and $w = y - 4$, so $v = w = 0$ at the critical point. To write

$V(x, y)$ in terms of v and w, we make the substitutions $x = v + 3$ and $y = w + 4$ into the definition of V:

$$\begin{aligned} V(v+3, w+4) &= 2(v+3)^2 + (w+4)^2 - 4(v+3) - 8((v+3)+(w+4)) \\ &= 2(v^2+6v+9) + (w^2+8w+16) - 4v - 12 - 8(v+w+7) \\ &= 2v^2 + 12v + 18 + w^2 + 8w + 16 - 4v - 12 - 8v - 8w - 56 \\ &= 2v^2 + w^2 + (12v - 4v - 8v) + (8w - 8w) + (18 + 16 - 12 - 56) \\ &= 2v^2 + w^2 - 34. \end{aligned}$$

The linear expressions (i.e., av and bw) entirely canceled out! This happened exactly because we are at a critical point. The quadratic part $2v^2 + w^2$ is visibly positive away from $v = w = 0$, so the minimum value is -34 and it is attained exactly at $v = w = 0$. But $v = w = 0$ is precisely the critical point P, so P is a global minimum for V (at which the minimal value is $V(3, 4) = -34$).

We got lucky because not only did the linear terms entirely disappear when we shifted the critical point to the origin, but also no cross-term (i.e., a vw-term) ever appeared. If there had been a cross-term then we wouldn't have been able to eye-ball the behavior at $v = w = 0$ as readily as we did. The presence of cross-terms in quadratic expressions turns out to be a pervasive issue in the determination of whether or not a critical point for a general function $f : \mathbf{R}^n \to \mathbf{R}$ is a local extremum (e.g., this issue arose in Example 10.2.7, where it was bypassed via cleverness), so we have to learn how to grapple with it. A systematic way to handle cross-terms will be discussed in Sections 26.3–26.4, once we have the appropriate further background in linear algebra. ■

10.3. Fitting curves to data. Let us now revisit the problem of fitting lines or curves to data. We systematically work through the computations and discuss how to interpret the answer.

Example 10.3.1. Suppose we are given a collection of data points (x_i, y_i), $i = 1, \ldots, n$, not all on the same vertical line; i.e., not all the x_i have the same value. (If they all have the same value, a vertical line fits the data precisely!) We want to find the value of (m, b) which minimizes

$$\mathrm{E}(m, b) = \sum_{i=1}^{n} (y_i - mx_i - b)^2.$$

We explained in Section 7.3 how to find the solution using linear algebra. Now we do it using calculus (i.e., partial derivatives) to see that the answers agree. We begin by computing the two partial derivatives: $\dfrac{\partial \mathrm{E}}{\partial b} = \sum_{i=1}^{n} -2(y_i - mx_i - b)$, $\dfrac{\partial \mathrm{E}}{\partial m} = \sum_{i=1}^{N} -2x_i(y_i - mx_i - b)$. The values of m and b we are looking for must satisfy $E_m(m, b) = 0$, $E_b(m, b) = 0$, so we need to solve the system of simultaneous linear equations

$$\sum_{i=1}^{n} (y_i - mx_i - b) = 0, \quad \sum_{i=1}^{n} x_i(y_i - mx_i - b) = 0.$$

It looks like a mess, but remember that each of x_i and y_i are numbers that we know beforehand. So we can rewrite this in a more transparent way as

$$m \sum_{i=1}^{n} x_i + nb = \sum_{i=1}^{n} y_i, \quad m \sum_{i=1}^{n} x_i^2 + b \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i y_i.$$

This is "2 equations in 2 unknowns" m and b (the coefficients involve expressions in the known data points (x_i, y_i) and n, so they are known numbers). To be concrete, for Example 7.3.2 these equations become

$$-15m + 5b = 0, \quad 55m - 15b = 12.$$

The first equation shows that $b = 3m$, and inserting this into the second equation gives $10m = 12$, so $m = 6/5$ and hence $b = 18/5$. Thus, the line which best fits this data is $y = (6/5)x + (18/5)$, as shown in Figure 10.3.1. This agrees with our earlier answer (as it must)!

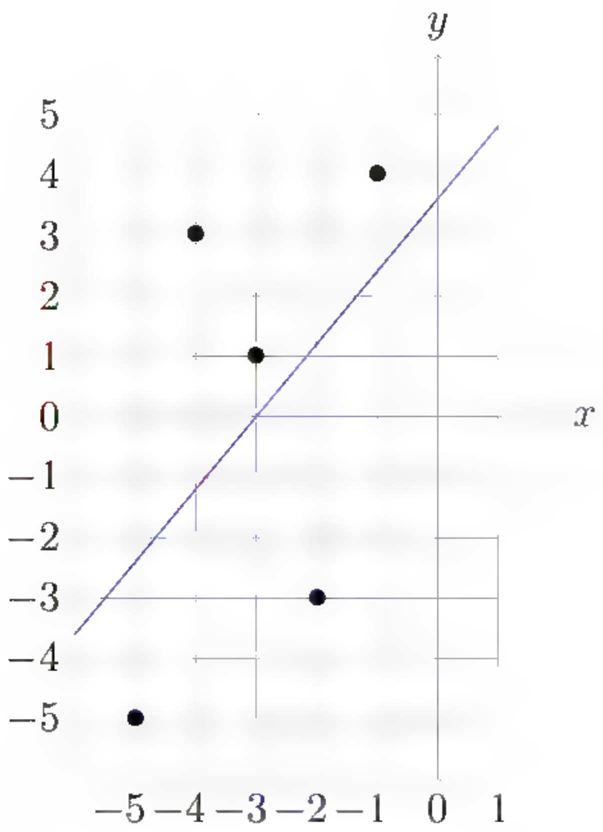


FIGURE 10.3.1. The plot for Example 7.3.2.

■

Example 10.3.2. Here is a closely related example which builds on the idea that sometimes we want to find more complicated curves which best fit a certain set of data. So now let us find constants A, B, C for which the function $y = A\sin(Bx + C)$ best fits data points $(x_1, y_1), \ldots, (x_N, y_N)$. When the data represents some phenomenon which we expect to be periodic (e.g., temperatures in a given location over a several-year period), it is quite reasonable to use the periodic sine function as a model for the change over the seasons.

We proceed just as before, by trying to find a minimum of the total squared error

$$E(A, B, C) = \sum_{i=1}^{N} (y_i - A\sin(Bx_i + C))^2 .$$

We compute the three partial derivatives $\partial E/\partial A$, $\partial E/\partial B$ and $\partial E/\partial C$, and this requires a bit more work than in the previous example. Please make sure you understand the following computation:

$$\frac{\partial E}{\partial A} = \sum_{i=1}^{N} -2\sin(Bx_i + C)\,(y_i - A\sin(Bx_i + C)),$$

$$\frac{\partial E}{\partial B} = \sum_{i=1}^{N} -2Ax_i\cos(Bx_i + C)\,(y_i - A\sin(Bx_i + C)),$$

$$\frac{\partial E}{\partial C} = \sum_{i=1}^{N} -2A\cos(Bx_i + C)\,(y_i - A\sin(Bx_i + C)).$$

■

This is an instructive example because this system of equations is pretty complicated and it is very unlikely that we can find the exact values of the solutions (A, B, C) in terms of the data points (x_i, y_i). For such a system of equations it is reasonable to try to think about how to find a numerical approximation to critical points (A, B, C).

10.4. Extrema on boundaries. In the preceding discussion, for $f : \mathbf{R}^n \to \mathbf{R}$ we have been looking for critical points (i.e., points $\mathbf{a} \in \mathbf{R}^n$ at which all partial derivatives $\partial f/\partial x_i$ vanish) since all local maxima and local minima for f have to be among such points. But in practical applications we often need to find maxima or minima of f when f is regarded as a function only on some region D in $\mathbf{R}^n$. Working on a fixed region often corresponds to practical constraints:

Example 10.4.1. The coordinates x_i may have a physical or other real-world meaning that constrains them to always be non-negative (e.g., height above the ground, or some temperature in Kelvin). In such cases we may use $D = \{(x_1, \ldots, x_n) \in \mathbf{R}^n : x_i \geq 0 \text{ for all } i\}$. ■

Example 10.4.2. In an economic model, x_i may correspond to the total value of the ith investment. Although negative monetary value in economics can be interpreted as debt, in the context of stock market investments there may be no real meaning for a negative value of an investment, so $x_i \geq 0$.

But there might also be some large possible value V_i of x_i beyond which we don't want to use the economic model anymore, so we may also want to demand $x_i \leq V_i$ for all i. Consequently, we may use $D = \{(x_1, \ldots, x_n) \in \mathbf{R}^n : 0 \leq x_i \leq V_i \text{ for all } i\}$. ■

When seeking maximal and minimal values of a function on a region D (rather than on the entirety of $\mathbf{R}^n$), extrema might be attained at points that are *not* among those where partial derivatives vanish. This is a phenomenon that arises in single-variable calculus, as we now review to illustrate the point.

Example 10.4.3. Consider the task of finding the maxima and minima of the function

$$f(x) = \tfrac{1}{3}x^3 - 3x^2 + 5x - 5$$

on the interval $[-2, 6]$ (see the blue curve in Figure 10.4.1).

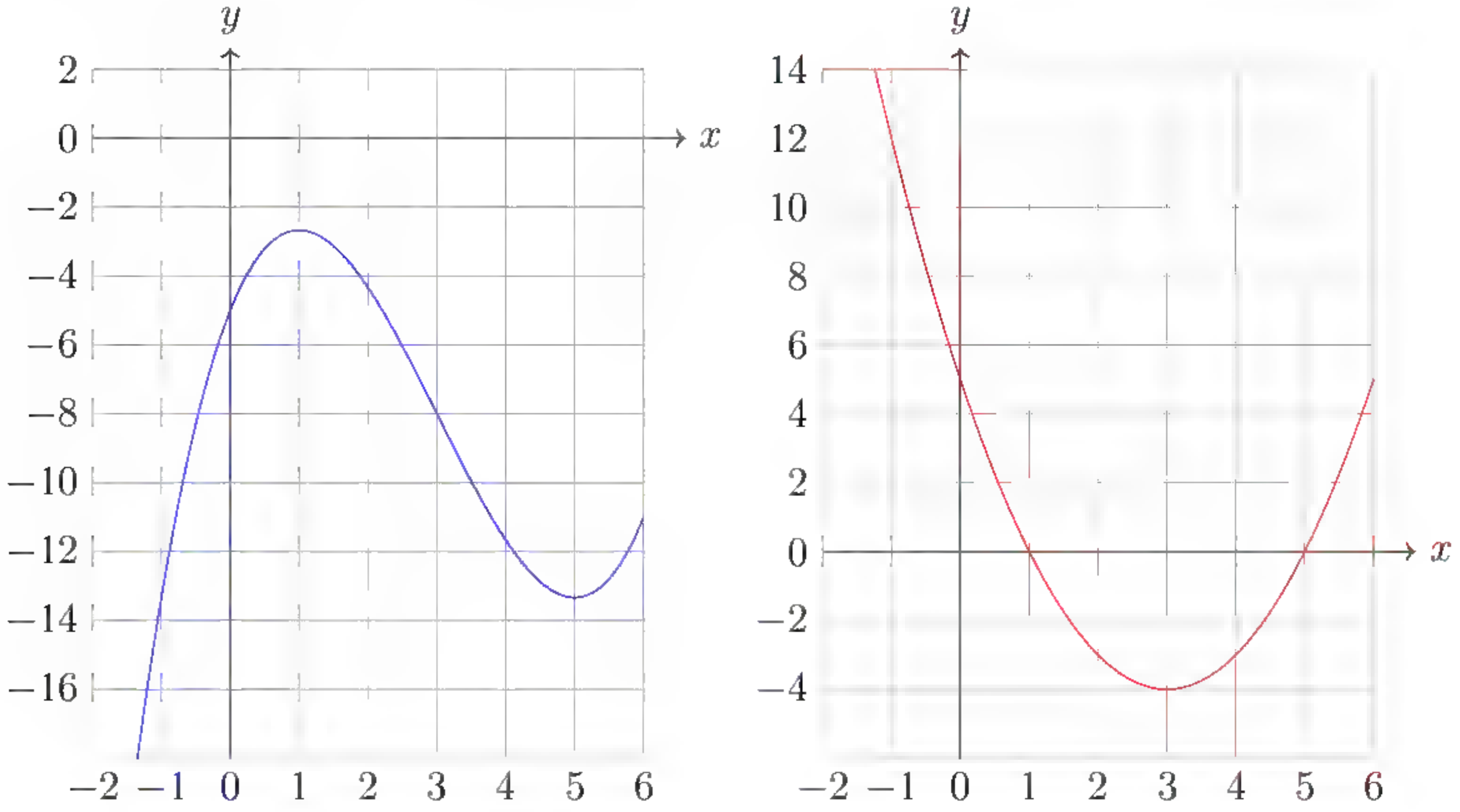


FIGURE 10.4.1. The graphs $y = f(x)$ (left) and $y = f'(x)$ (right).

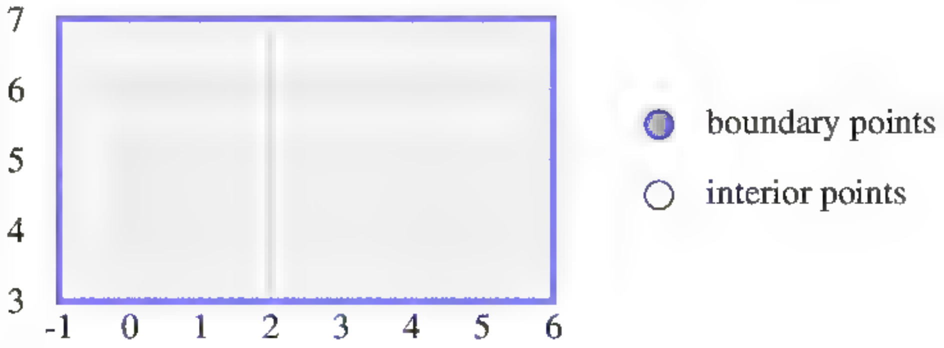
Evaluating f at its two critical points yields $f(1) = -8/3 = -2.666\ldots$ and $f(5) = -40/3 = -13.333\ldots$. Hence, without any further inspection, one might guess that $-40/3$ is the minimum and $-8/3$ is the maximum for f on the interval $[-2, 6]$. However, this is not the case because we have to always **remember to check what happens at the endpoints**! In fact, the graph of f as the blue curve in Figure 10.4.1 shows quite visibly that an extremum on a closed interval (such as on $[-2, 6]$ for us) can occur at an endpoint without a horizontal tangent line there.

Explicitly, we compute $f(-2) = -89/3 = -29.666\ldots$ and $f(6) = -11$. Thus, amongst the values of f at its two critical points $x = 1$ and $x = 5$ as well as at the two endpoints $x = -2$ and $x = 6$, the smallest value is $f(-2) = -89/3$ and the largest is $f(1) = -8/3$. In other words, the maximum on this interval occurs at $x = 1$, which lies in the open interval $(-2, 6)$ (i.e., away from the endpoints) whereas the minimum occurs at the endpoint $x = -2$. ■

The moral of Example 10.4.3 is to remind us that in single-variable calculus, a local extremum $x = c$ for a function $f : I \to \mathbf{R}$ on an interval I is necessarily a critical point *only* when c is not an endpoint of I. It can happen (as in Example 10.4.3) that an endpoint of I is a local extremum for f on the interval I, and in such cases that endpoint need *not* be a point where f' vanishes.

It is natural to expect an analogue of the above "endpoint" phenomenon (i.e., local extremum on a region occurring at a non-critical point) may arise in the multivariable setting when studying $f : D \to \mathbf{R}$ for a subset D of $\mathbf{R}^n$. The replacement of "endpoint" for such D is called a *boundary* point. Rather than delve into general concepts, we will limit ourselves to some specific examples that can be readily visualized and convey the main idea.

Example 10.4.4. Let D be the rectangle $\{(x, y) : -1 \leq x \leq 6,\ 3 \leq y \leq 7\}$. The points along the outer edge are called *boundary* points, and the other points of D are called its *interior* points.



Explicitly, the boundary points are those points in D of the form: $(-1, y)$ (left edge), $(6, y)$ (right edge), $(x, 3)$ (bottom edge), and $(x, 7)$ (top edge). The interior points are those (x, y) satisfying both of the strict inequalities

$$-1 < x < 6 \text{ and } 3 < y < 7.$$

■

Example 10.4.5. Let D be the solid ball

$$\{(x, y, z) \in \mathbf{R}^3 : (x-3)^2 + (y+2)^2 + (z-5)^2 \leq 9 = 3^2\}$$

of radius 3 centered at $(3, -2, 5)$. In this case the points of the outer sphere

$$\{(x, y, z) \in \mathbf{R}^3 : (x-3)^2 + (y+2)^2 + (z-5)^2 = 9\}$$

are called the *boundary* points, and the other points of D are called its *interior* points. Explicitly, the interior points are those $(x, y, z) \in \mathbf{R}^3$ whose distance to the center of the ball is strictly less than 3:

$$(x-3)^2 + (y+2)^2 + (z-5)^2 < 9.$$

■

Hopefully from the above two examples you can imagine the appropriate notions of boundary point and interior point for other familiar regions in a plane (e.g., the region bounded by a polygon) or in 3-dimensional space (e.g., the region bounded by a cylinder with finite height). In Section 10.5 we define the notions of "boundary point" and "interior point" more precisely, for those who are interested. The analogue in $\mathbf{R}^n$ with $n > 1$ of treating endpoints of an interval separately when checking for extrema is:

Theorem 10.4.6. For a function $f : \mathbf{R}^n \to \mathbf{R}$ and a region D inside $\mathbf{R}^n$, suppose the function $f : D \to \mathbf{R}$ considered on D has a local extremum at a point $\mathbf{a} \in D$ (i.e., $f(\mathbf{x}) \geq f(\mathbf{a})$ for all $\mathbf{x} \in D$ near $\mathbf{a}$, or $f(\mathbf{x}) \leq f(\mathbf{a})$ for all $\mathbf{x} \in D$ near $\mathbf{a}$).

The point $\mathbf{a}$ must be a critical point of f when $\mathbf{a}$ is in the interior of D (but generally not otherwise). In particular, any local extremum of $f : D \to \mathbf{R}$ either is a critical point on the interior of D or is a boundary point of D.

In Theorem 10.4.6, the notion of "local extremum" is meant from the perspective of evaluating f at points of D near $\mathbf{a}$; it is *not* assumed that the values of f at points near $\mathbf{a}$ off of D have any relation to $f(\mathbf{a})$. This is a "constrained" sense of optimization (a constrained local maximum or minimum).

Strictly speaking, in Theorem 10.4.6 we need to assume f is "differentiable" in an appropriate n-variable sense, the same issue which we discussed in Remark 10.2.4.

A huge difference between the study of local extrema at boundary points of regions D in $\mathbf{R}^n$ when $n > 1$ as opposed to when $n = 1$ is that the possibilities for the geometry of the boundary of D are much richer when $n > 1$ (even when $n = 2$) than when $n = 1$. For example, a closed disk has boundary a circle whereas a closed (bounded) interval in $\mathbf{R}$ has boundary that is just a pair of points. This makes the study of local extrema of $f : D \to \mathbf{R}$ at boundary points much richer when $n > 1$ (and the task will be taken up systematically in Chapter 12 using a new concept that pervades optimization problems in economics, physics, and many other fields: Lagrange multipliers).

Example 10.4.7. To illustrate the principle in Theorem 10.4.6, we first revisit the rectangle

$$D = \{(x, y) \in \mathbf{R}^2 : -1 \leq x \leq 6, 3 \leq y \leq 7\}$$

from Example 10.4.4. For $f(x, y)$ defined on this rectangle, suppose we wish to locate the points where $f : D \to \mathbf{R}$ attains maximal and minimal values. (We will consider an explicit f on another explicit rectangle in Example 10.4.8 below, but for now we want to describe the general method to be carried out.)

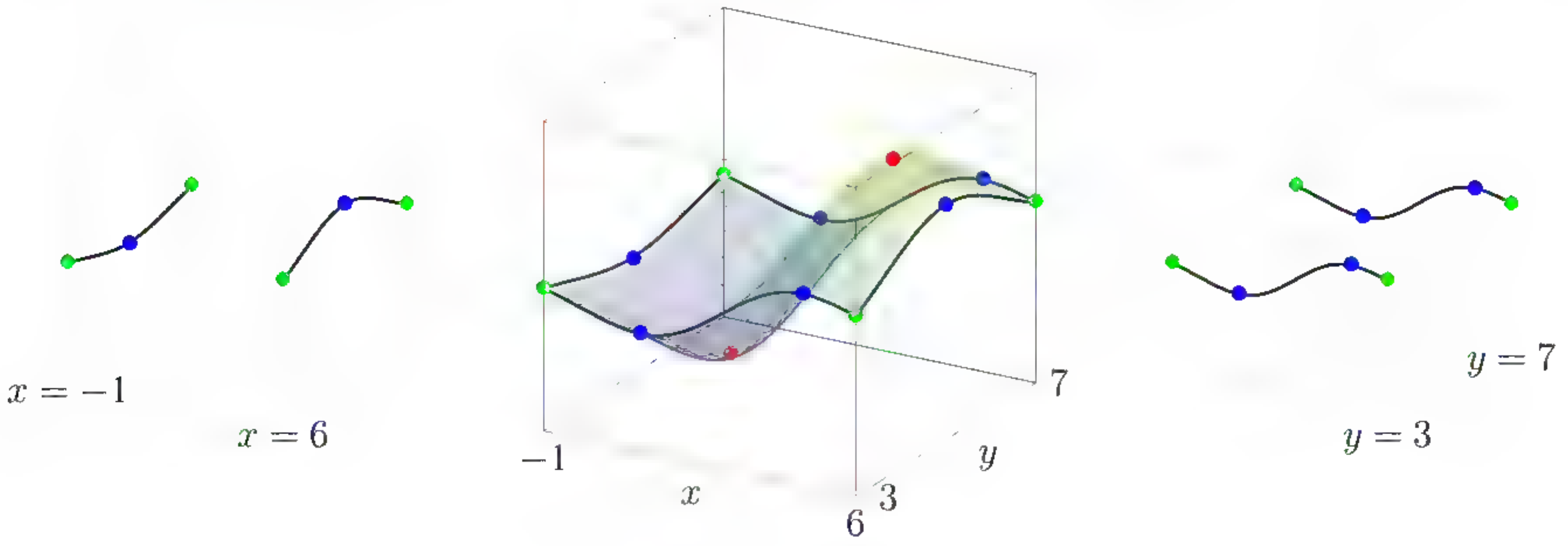


FIGURE 10.4.2. A variety of potential local extrema for a function on a rectangle

To aid in the discussion, in the middle of Figure 10.4.2 we show the graph in $\mathbf{R}^3$ of some function $f(x, y)$ over the region D in the xy-plane. As a first step, we seek points where both partial derivatives f_x and f_y vanish. There might be several points in D at which both such partial derivatives vanish. For example, suppose $(0, 5)$ and $(5, 6)$ are the only points where f_x and f_y both vanish, shown as red points in Figure 10.4.2. Does f necessarily achieve its maximal and minimal values on D at those two points? In general, not necessarily! We must also examine the values of f at the boundary points of D (i.e., along the edges of the rectangle).

That is, we should extremize f along each side, and compare those values against $f(0, 5)$ and $f(5, 6)$. More explicitly, considering f on each side of the rectangle gives us one-variable optimization problems:

(i) The left edge consists of points $(-1, y)$ for $3 \leq y \leq 7$, so looking at f on the left edge corresponds to the function $f(-1, y)$ on the interval $[3, 7]$.
(ii) The right edge consists of points $(6, y)$ for $3 \leq y \leq 7$, so looking at f on the right edge corresponds to the function $f(6, y)$ on the interval $[3, 7]$.
(iii) The top edge consists of points $(x, 7)$ for $-1 \leq x \leq 6$, so looking at f on the top edge corresponds to the function $f(x, 7)$ on the interval $[-1, 6]$.
(iv) The bottom edge consists of points $(x, 3)$ for $-1 \leq x \leq 6$, so looking at f on the bottom edge corresponds to the function $f(x, 3)$ on the interval $[-1, 6]$.

To summarize, finding extrema of f along the boundary corresponds to finding extrema of the four single-variable functions

$$f(x, 7), \quad f(x, 3), \quad f(-1, y), \quad f(6, y) \tag{10.4.1}$$

with $-1 \leq x \leq 6$ in the first two cases and $3 \leq y \leq 7$ in the last two cases. We can handle these via single-variable calculus. The graphs of these one-variable functions are shown as planar slices on the left and right in Figure 10.4.2.

For instance, finding critical points of the third of the four functions in (10.4.1) away from the endpoints of its interval $[3, 7]$ corresponds to analyzing the equation $f_y(-1, y) = 0$ on the interval $(3, 7)$, and similarly with $f_x(x, 7) = 0$ on the open interval $(-1, 6)$ for the first function in (10.4.1), and so on. Solutions to these conditions are additional candidates for extrema of f on D. These points are shown as blue points in Figure 10.4.2.

Finally, there are also the endpoints of each of these edge intervals, corresponding to the 4 corners of the rectangle D: $(-1, 3), (-1, 7), (6, 3), (6, 7)$. These points are shown as green points in Figure 10.4.2.

■

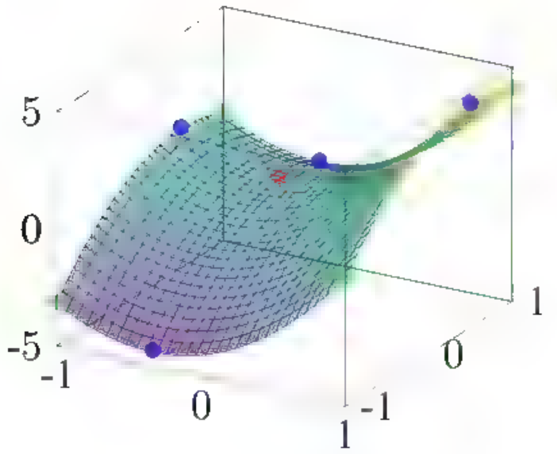
Example 10.4.8. Here is a case with a specific f on a specific region in $\mathbf{R}^2$ to illustrate a complete calculation in practice. Consider the function

$$f(x, y) = 3x^2 - 2y^2 + 2x + 2y - 1$$

on the rectangle

$$D = \{(x, y) \in \mathbf{R}^2 : -1 \leq x \leq 1, -1 \leq y \leq 1\}.$$

Its graph is shown in Figure 10.4.3, but we will argue in a graph-free way that can be adapted to the n-variable setting for any n.



The graph of $f(x,y) = 3x^2 - 2y^2 + 2x + 2y - 1$.

The interior critical point is shown in red.

The edge critical points are blue.

The corner points are green.

The extrema can only occur at these colored points.

FIGURE 10.4.3. Potential local extrema on the interior and boundary of the region D

We first compute

$$f_x = 6x + 2, \quad f_y = -4y + 2,$$

so

$$f_x = 0 \text{ implies } x = -1/3, \quad f_y = 0 \text{ implies } y = 1/2.$$

Hence, the vanishing of *both* partial derivatives happens precisely at the single point $(-1/3, 1/2)$ (which we observe is on the interior of the rectangle D). Note that $f(-1/3, 1/2) = -5/6$. This single critical point in the interior is shown in red in the figure.

Next, we analyze how f behaves on the edges of D. Focusing on the bottom edge amounts to studying the function $f(x,-1) = 3x^2 + 2x - 5$ on the interval $[-1, 1]$. This has derivative $f_x(x,-1) = 6x+2$, which vanishes at precisely $x = -1/3$, so $(-1/3, -1)$ is another candidate. We evaluate $f(-1/3, -1) = -16/3$.

In a similar way along the other edges, we compute:

(i) $f_x(x, 1) = 0$ at precisely $x = -1/3$ (with $f(-1/3, 1) = -4/3$),
(ii) $f_y(-1, y) = 0$ at precisely $y = 1/2$ (with $f(-1, 1/2) = 1/2$),
(iii) $f_y(1, y) = 0$ at precisely $y = 1/2$ (with $f(1, 1/2) = 9/2$).

These four critical points "on the edges" are shown in blue.

Finally, evaluating at the 4 corners, we have

$$f(-1,-1) = -4, \quad f(-1,1) = 0, \quad f(1,-1) = 0, \quad f(1,1) = 4.$$

These points are shown on the graph in green.

Comparing the values of f at all of these points (the one critical point on the interior, the one critical point on each edge away from the corners, and all 4 corners) and seeing which is biggest and which is smallest, we see that f has its global maximum on the unit square at the boundary point $(1, 1/2)$, with $f(1, 1/2) = 9/2$, and its global minimum at the boundary point $(-1/3, -1)$, with $f(-1/3, -1) = -16/3$.

This method does *not* help at all in determining whether the other points that we identified above are at least *local* maxima or minima over the domain, or if the critical point in the interior is a saddle point. ■

The general principle for functions $f : D \to \mathbf{R}$ on regions D in $\mathbf{R}^2$ is that to find extrema we have to really consider two or more problems (generalizing the experience in single-variable calculus of treating endpoints separately for extrema problems).

(i) First we must identify the *possible* extrema in the interior, which we do using the first partial derivative test (to find critical points there): Theorem 10.4.6.
(ii) Then we must look at the restriction of the function to boundary curves. These are 1-dimensional, so in practice the problem there eventually reduces to an extrema problem for

a function of one variable (by parameterizing each boundary curve), which we hope in turn to solve using our knowledge of single-variable calculus.

Moving beyond 2 variables to the 3-variable case, if we are instead considering a function $f : D \to \mathbf{R}$ for a region D inside $\mathbf{R}^3$ such as a solid cube, then we first consider its extrema in the interior (which must be among the critical points, by Theorem 10.4.6). Then we need to find extrema on each of the square faces (bringing us back to an instance of our 2-dimensional extrema problem on a region with boundaries). In particular, the study of these square faces requires separate work on its edges and in particular at all of the vertices of the cube. As you can see, this is potentially a lot of work!

We hope that the preceding examples convey a sense of how one can aim to find maxima and minima in any given situation on a reasonable region D inside $\mathbf{R}^n$ by combining the technique of critical points on the interior and "lower-dimensional" extrema problems along the boundary.

Remark 10.4.9. The problem of finding extrema of functions is one of the most important applications of multivariable calculus, and arises in many contexts (economics with maximizing utility functions, machine learning with gradient descent for minimization, physical sciences with energy minimization, SVD with Rayleigh quotient maximization, etc.). Often the functions that one encounters "in nature" are complicated, or may not be given by an elementary formula or by any explicit rule (e.g., problems in computational biochemistry and computer science are usually of this type). In such cases, one often investigates extrema questions numerically.

The general principle we have learned is that maxima and minima at interior points occur where the partial derivatives vanish simultaneously. However, the task of finding critical points often must also be carried out numerically, especially when the function of interest is not given by any explicit formula. There are extremely efficient and quick algorithms for locating to within an arbitrary degree of precision the critical points of f. An indication of one way this is done is given in Appendix I.

10.5. Details on interior and boundary, and relation of extrema to critical points. The moral of Example 10.4.3 is that we could not detect via *derivatives* that the minimum of f occurs at the endpoint -2 because when we only consider values $-2+h$ which lie inside the interval; i.e., small $h \geq 0$, then $f(-2+h) \geq f(-2)$, but this is not enough information to conclude that the limiting difference quotient

$$\lim_{h \to 0} \frac{f(-2+h) - f(-2)}{h}$$

must vanish. Indeed, for that we would have also needed to know that $f(-2+h) \geq f(-2)$ with small $h < 0$, which is false if we consider the values of f outside the interval. More specifically, it does not even make sense to consider $f(-2+h)$ with $h < 0$ if we *only allow ourselves* to consider f as a function on $[-2, 6]$.

The desire to extend the separate treatment of endpoints of intervals to the setting of extrema problems in $\mathbf{R}^n$ for $n > 1$ leads us to define what it means for a point a to be an *interior point* of a region D inside $\mathbf{R}^n$, as follows.

Definition 10.5.1. A point $\mathbf{a} \in D$ is an *interior* point if *all* nearby points $\mathbf{x} \in \mathbf{R}^n$ also lie in D. Otherwise we call $\mathbf{a}$ a *boundary* point of D.

So in the 1-dimensional case with $D = [-2, 6]$, any point a with $-2 < a < 6$ is an interior point: any value $a + h$ with h sufficiently small still lies inside the interval $[-2, 6]$. However, if $a = -2$ then $a + h = -2 + h$ lies in the interval D only when h is sufficiently small and positive: all points $-2 + h$ with $h < 0$ lie outside the interval.

Similar reasoning with h of the opposite sign holds at the other endpoint $a = 6$. We conclude that $a = -2$ and $a = 6$ are the boundary points of $[-2, 6]$ (as we would want to be the case) since they fail the criterion for being interior points. In other words, for a closed bounded interval $[b, c]$ with $b < c$, the interior points are those in the open interval (b, c) whereas the boundary points are b and c.

The situation in $\mathbf{R}^n$ with $n > 1$ is a bit more complicated for boundary possibilities, so we are just going to consider some examples that should give you the basic idea.

Example 10.5.2. Let D be the rectangle $\{(x, y) : -2 < x < 6,\ 3 < y < 7\}$ omitting the edge segments (where $x = -2$ or 6, and where $y = 3$ or 7). We claim that every point $\mathbf{a} = (a_1, a_2) \in D$ is an interior point. The reason is that, by definition, $-2 < a_1 < 6$, $3 < a_2 < 7$, so for any $\mathbf{a} + \mathbf{h} = (a_1 + h_1, a_2 + h_2)$ with h_1, h_2 sufficiently near 0 (depending on how far the a_i's are from the endpoints of the respective intervals in the two coordinate directions) we see visually that $\mathbf{a} + \mathbf{h}$ still lies in D.

This situation is to be contrasted with the related region $D' = \{(x, y) : -2 \le x \le 6,\ 3 \le y \le 7\}$ for which we have included the edge segments. For any point $\mathbf{a}$ on those edges, $\mathbf{a} + \mathbf{h}$ lies outside D' with $\mathbf{h}$ pointing in a suitable direction (depending on $\mathbf{a}$ and as small as we please). As an example, with $\mathbf{a} = (-2, 4)$ the point $\mathbf{a} + \mathbf{h}$ never belongs to D' when $h_1 < 0$, no matter how small we make $\mathbf{h}$. ■

Example 10.5.3. Here is an example to check your understanding of Definition 10.5.1. Let D be the unit disc

$$\{\mathbf{x} = (x_1, x_2) \in \mathbf{R}^2 : x_1^2 + x_2^2 < 1\},$$

omitting the outer circle of radius 1. Check for yourself by drawing pictures that if $\mathbf{a}$ is any point in D and $\mathbf{h}$ is a short enough vector (in fact, we may take $\mathbf{h}$ to be any vector with $\|\mathbf{h}\| < 1 - \|\mathbf{a}\|$) then $\mathbf{a} + \mathbf{h} \in D$.

On the other hand, for the related region $D' = \{\mathbf{x} : x_1^2 + x_2^2 \le 1\}$ that includes the outer circle, a point $\mathbf{a}$ on that outer circle is not on the interior. Indeed, there are many nonzero vectors $\mathbf{h}$ with positive length as short as we wish for which $\mathbf{a} + \mathbf{h}$ lies outside D' (i.e., has length larger than 1): use such $\mathbf{h}$ pointing in the same direction as $\mathbf{a}$. ■

The key takeaway from this discussion is the following. If f is defined on some region D inside $\mathbf{R}^n$, and if $\mathbf{a}$ is an interior point of D then the difference quotient definition of the jth partial derivative of f at $\mathbf{a}$ makes good sense – we can evaluate

$$f(a_1, \dots, a_j + h_j, \dots, a_n)$$

for all h_j sufficiently near 0, both *positive and negative* values are allowed – and furthermore, if $f : D \to \mathbf{R}$ attains a local maximum or local minimum at an interior point $\mathbf{a} \in D$ then all such partial derivatives vanish at $\mathbf{a}$. On the other hand, if $\mathbf{a}$ is not an interior point of D then although we might still be able to calculate a partial derivative at that point by taking a limit of a difference quotient

$$\lim_{h_j \to 0} \frac{f(a_1, \dots, a_j + h_j, \dots, a_n) - f(\mathbf{a})}{h_j}$$

with $h_j \to 0$ only from the positive side or only from the negative side (a "one-sided limit"), this is *not enough* to conclude that the partial derivatives vanish when $f : D \to \mathbf{R}$ has a local maximum or local minimum at such a boundary point $\mathbf{a}$.

Chapter 10 highlights (links to highlights in and next chapters)

Notation	**Meaning**	**Location in text**
nothing new!		

Concept	**Meaning**	**Location in text**
local max. and local min. for $f : \mathbf{R}^n \to \mathbf{R}$	point $\mathbf{a} \in \mathbf{R}^n$ so that for all $\mathbf{x}$ near $\mathbf{a}$ either always $f(\mathbf{x}) \leq f(\mathbf{a})$ (local maximum) or always $f(\mathbf{x}) \geq f(\mathbf{a})$ (local minimum)	Definition 10.2.1
critical pt for $f : \mathbf{R}^n \to \mathbf{R}$	point $\mathbf{a} \in \mathbf{R}^n$ at which all partial derivatives of f vanish	Definition 10.2.3
saddle pt for $f : \mathbf{R}^n \to \mathbf{R}$	point $\mathbf{a} \in \mathbf{R}^n$ so that there are different lines L and L' through $\mathbf{a}$ with f restricted to L having a local maximum at $\mathbf{a}$ and f restricted to L' having a local minimum at $\mathbf{a}$	Definition 10.2.10
local extrema for $f : D \to \mathbf{R}$ on region D in $\mathbf{R}^n$	point $\mathbf{a} \in D$ so that for all $\mathbf{x} \in D$ near $\mathbf{a}$ either always $f(\mathbf{x}) \leq f(\mathbf{a})$ (local max.) or always $f(\mathbf{x}) \geq f(\mathbf{a})$ (local min.)	in statement of Theorem 10.4.6

Result	**Meaning**	**Location in text**
local extremum for $f : \mathbf{R}^n \to \mathbf{R}$ is critical point	if $\mathbf{a} \in \mathbf{R}^n$ is a local maximum or local minimum for f then $f_{x_i}(\mathbf{a}) = 0$ for all i	Theorem 10.2.2
local extremum for $f : D \to \mathbf{R}$ is either critical pt in interior or is boundary pt	for a region D in $\mathbf{R}^n$, if $\mathbf{a} \in D$ is a local maximum or local minimum for $f : D \to \mathbf{R}$ then either $\mathbf{a}$ is a boundary point of D or else it is in the interior of D with all $f_{x_i}(\mathbf{a})$ equal to 0	Theorem 10.4.6

Skill	**Location in text**
find critical points as *candidates* for local extrema of $f : \mathbf{R}^n \to \mathbf{R}$	Examples 10.2.6, 10.2.7
use contour plot to identify local extrema and saddle points for a function $f : \mathbf{R}^2 \to \mathbf{R}$	Example 10.2.13
find maxima and minima for $f : D \to \mathbf{R}$ on region D by treating boundary points separately (may not be critical points)	Examples 10.4.7, 10.4.8

11. Gradients, local approximations, and gradient descent

For a scalar-valued function $f : \mathbf{R}^n \to \mathbf{R}$, we now package all of its partial derivatives into a single *vector-valued* function denoted $\nabla f : \mathbf{R}^n \to \mathbf{R}^n$ called the *gradient* of f. We use the gradient to compute a good local approximation to a scalar-valued function $f : \mathbf{R}^n \to \mathbf{R}$, and discuss how to visualize it.

By the end of this chapter, for a function $f : \mathbf{R}^n \to \mathbf{R}$ and point $\mathbf{a} \in \mathbf{R}^n$, you should be able to:

- use partial derivatives to compute the gradient $(\nabla f)(\mathbf{a}) \in \mathbf{R}^n$ at $\mathbf{a}$ (it is a vector that packages all n partial derivatives of f at $\mathbf{a}$);
- compute a local approximation to f near $\mathbf{a} \in \mathbf{R}^n$ using $(\nabla f)(\mathbf{a})$;
- use ∇f to compute the equation of the tangent plane (resp. tangent line) to a level set $f = c$ when $n = 3$ (resp. $n = 2$);
- identify directions of most rapid increase or decrease, symbolically or (for $n = 2$) on a plot.

11.1. The linear approximation for a scalar-valued function. For a function f of a single variable x, we know that a small change h in the value of x near $x = a$ causes an approximate change of $f'(a)h$ in the value of $f(x)$ near $x = a$: $f(a+h) \approx f(a) + f'(a)h$ for all h near 0. We can write this in another way: for x very close to a,

$$f(x) \approx f(a) + f'(a)(x-a), \tag{11.1.1}$$

which is the same statement of approximation except with x in place of $a + h$ (so $x - a = h$ is small). Now the right side is the equation of a line: the tangent line to the graph of $y = f(x)$ at the point $(a, f(a))$.

What is the analogue for a scalar-valued function $f : \mathbf{R}^n \to \mathbf{R}$?

Consider $f : \mathbf{R}^n \to \mathbf{R}$. We will sometimes write the inputs to f as vectors; e.g., when $n = 2$ we think of f as a function $f\left(\begin{bmatrix} x \\ y \end{bmatrix}\right)$.

Definition 11.1.1. The *gradient of f* is defined to be

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}.$$

Note that the gradient of f is a vector-valued function $\mathbf{R}^n \to \mathbf{R}^n$: its value $(\nabla f)(\mathbf{a})$ at $\mathbf{a} \in \mathbf{R}^n$ is an n-vector. For $\mathbf{x}$ near $\mathbf{a} \in \mathbf{R}^n$, the *linear approximation* to f is

$$f(\mathbf{x}) \approx f(\mathbf{a}) + ((\nabla f)(\mathbf{a})) \cdot (\mathbf{x} - \mathbf{a}). \tag{11.1.2}$$

Observe that this looks just like the single-variable case (11.1.1) except that now there are vectors and a dot product involved. (When $n = 1$ this recovers exactly the familiar single-variable case, with $(\nabla f)(x)$ equal to the 1-vector $[f'(x)] \in \mathbf{R}^1 = \mathbf{R}$.)

Although we encourage you to remember the vector-language formulation (11.1.2) since it looks so similar to the single-variable case, let us write it out explicitly in the case of a two-variable function (i.e., $n = 2$) without the vector notation. The formula says:

$$f(x,y) \approx f(a,b) + \underbrace{f_x(a,b)\,(x-a) + f_y(a,b)\,(y-b)}_{(\nabla f)(\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]) \left[\begin{smallmatrix} x & a \\ y & b \end{smallmatrix}\right]} \text{ for } (x,y) \text{ near } (a,b). \tag{11.1.3}$$

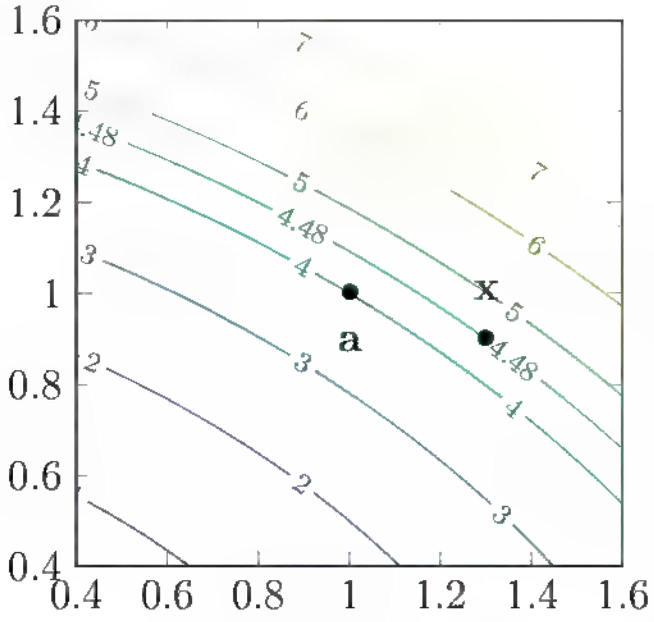
For an explanation of why (11.1.3) is true (applying equally well to the more general (11.1.2)), see Section 11.4. One thing that comes out of this explanation is that each individual term of (11.1.3) has a concrete meaning:

$$f(x,y) \approx f(a,b) \;+\; \underbrace{f_x(a,b)\,(x-a)}_{\text{change due to changing } x} \;+\; \underbrace{f_y(a,b)\,(y-b)}_{\text{change due to changing } y}$$

That is, the terms correspond to the change in f when we change *one variable at a time*.

We refer to either (11.1.2) or (11.1.3) as the *local approximation for f at* $\mathbf{a}$ or as the *linear approximation for f at* $\mathbf{a}$.

Example 11.1.2. Consider the function $f(x,y) = x^2 + 2y^2 + xy$ near the point $\mathbf{a} = (1,1)$. The contour plot is shown below.



A contour plot of $f(x,y) = x^2 + 2y^2 + xy$.
$\mathbf{a} = (1,1)$.
$\mathbf{x} = (1.3, 0.9)$.

FIGURE 11.1.1. A situation for using linear approximation to estimate $f(\mathbf{x})$

Working at the point $\mathbf{x} = (1.3, 0.9)$ near $\mathbf{a}$ as in Figure 11.1.1, let's estimate $f(1.3, 0.9)$ using the linear approximation formula with $\mathbf{a} = (1,1)$. That is, what is the f-value for the level set of f through $\mathbf{x} = (1.3, 0.9)$?

First we find $(\nabla f)(1,1)$:

$$\frac{\partial f}{\partial x} = 2x + y \quad \text{and} \quad \frac{\partial f}{\partial y} = 4y + x,$$

so $(\nabla f)(x,y) = \begin{bmatrix} 2x+y \\ 4y+x \end{bmatrix}$ and therefore $(\nabla f)(1,1) = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$. By (11.1.2),

$$\begin{aligned} f(1.3, 0.9) &\approx f(1,1) + (\nabla f)(1,1) \cdot \begin{bmatrix} .3 \\ -.1 \end{bmatrix} \\ &= 4 + \begin{bmatrix} 3 \\ 5 \end{bmatrix} \cdot \begin{bmatrix} .3 \\ -.1 \end{bmatrix} \\ &= 4.4. \end{aligned}$$

The exact value is $f(1.3, 0.9) = 4.48$, as shown in Figure 11.1.1. ■

Example 11.1.3. Let's approximate $f(1.03, 0.99)$ and $f(1.003, 0.999)$ for f as in the previous example.

Note that

$$\begin{aligned} f(1+h,1+k) &\approx f(1,1) + (\nabla f)(1,1)\cdot \begin{bmatrix} h \\ k \end{bmatrix} \\ &= 4 + \begin{bmatrix} 3 \\ 5 \end{bmatrix} \cdot \begin{bmatrix} h \\ k \end{bmatrix} \\ &= 4 + 3h + 5k. \end{aligned}$$

Consequently,

$$f(1.03, 0.99) \approx 4 + 3(0.03) + 5(-0.01) = 4.04 \quad f(1.003, 0.999) \approx 4 + 3(0.003) + 5(-0.001) = 4.004.$$

The exact values are $f(1.03, 0.99) = 4.0408$ and $f(1.003, 0.999) = 4.004008$. Note that the approximation is very accurate when h and k are very small. You should *not* expect the linear approximation $4 + 3h + 5k$ to be accurate for large h and k. For example, using the linear approximation with $h = k = 9$ to estimate $f(10, 10)$ gives 76 but $f(10, 10) = 400$. The huge deviation is visible in Figure 11.1.2, showing the graph and the plane $z = 4 + 3(x-1) + 5(y-1) = -4 + 3x + 5y$ giving the linear approximation to f at $(1, 1)$.

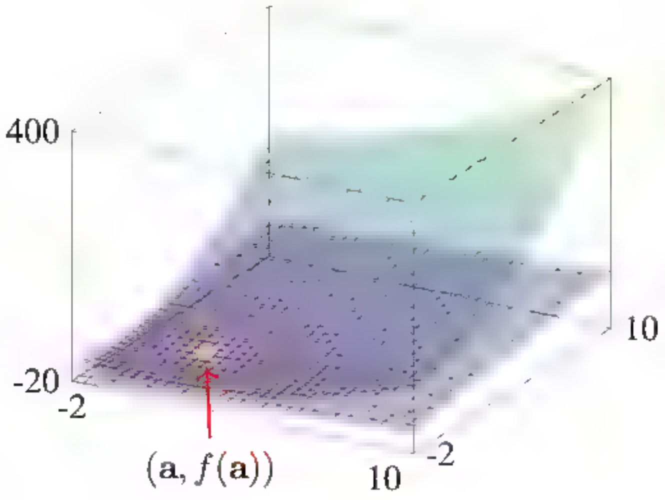


FIGURE 11.1.2. Comparing the graph of f and the linear approximation to f at $\mathbf{a} = (1, 1)$.

■

Example 11.1.4. The amount of oxygen that can dissolve in water depends substantially on temperature and air pressure, and, to a lesser extent, on salinity. Consider the following information (from [Col, p. 88]) at an air pressure of 1 atmosphere (the air pressure at ground level, corresponding to 760 mm of mercury (Hg)).

- The amount of dissolved oxygen in pure water at 20° C is 9.09 mg/L.
- The amount of dissolved oxygen in pure water at 25° C is 8.26 mg/L.
- The amount of dissolved oxygen at 20° C in water with salinity 1% is 8.57 mg/L.

Based on this information, let's estimate the amount of dissolved oxygen in water at $18°$ C with a salinity of 3.5% (approximately the salinity of seawater), still at an air pressure of 1 atmosphere. (Salinity is often measured in units of grams of salt per kilogram of water, called "parts per thousand" and denoted ppt; we will disregard such a 1000-fold scaling of the salinity number.)

Let $f(T, S)$ be the solubility of oxygen in water, measured in mg/L, at temperature T (in degrees Celsius) and salinity S. The above given information says

$$f(20, 0) = 9.09, \quad f(25, 0) = 8.26, \quad f(20, 0.01) = 8.57.$$

This lets us roughly estimate the partial derivatives via their definition in terms of difference quotients:

$$\frac{\partial f}{\partial T}(20,0) \approx \frac{f(25,0)-f(20,0)}{5} = -0.166, \quad \frac{\partial f}{\partial S}(20,0) \approx \frac{f(20,0.01)-f(20,0)}{0.01} = -52.$$

From this, get an estimate (with air pressure equal to 1 atmosphere)

$$f(T,S) \approx 9.09 + \begin{bmatrix} -0.166 \\ -52 \end{bmatrix} \cdot \begin{bmatrix} T-20 \\ S \end{bmatrix} = 9.09 - (0.166)(T-20) - 52S \tag{11.1.4}$$

for T near 20 and S near 0. (This estimate decreases as the salinity S increases, as oxygen solubility should.) We thereby obtain the numerical estimate

$$f(18, 0.035) \approx 9.09 - (0.166)(-2) - (52)(0.035) \approx 7.60$$

The real value is 7.68 mg/L, so the approximation worked quite well!

Likewise, if we keep T at 20 and try $S = 0.04$ and again take the air pressure to be 1 atmosphere then (11.1.4) gives the estimate

$$f(20, 0.04) \approx 9.09 - (52)(0.04) = 7.01$$

whereas the true value is 7.18 mg/L; still not bad! ■

Using just three measured values, we can estimate the solubility of oxygen in water at *any* other value of (T,S) that is *reasonably close* to $(20,0)$. As a general rule of thumb, one should never use approximate partial derivative information at some (a,b) to estimate $f(x,y)$ for (x,y) too far away from (a,b). This is for much the same reason that the "tangent line approximation" to a graph in single-variable calculus should never be used too far from the point at which the tangent line is based.

It is a bit surprising that the above estimate worked well even though we estimated $(\partial f/\partial T)(20,0)$ using a difference quotient involving a temperature value 5 units away from 20 (namely, we used $f(25,0)$). Next, we push our luck against the rule of thumb by trying a temperature value 5 units below 20 rather than just 2 units below 20 (going away from 20 in the direction opposite from $T = 25$ that was part of the input data).

Example 11.1.5. Let's use (11.1.4) to estimate the solubility of oxygen at a temperature of 15° C for $S = 0$ (pure water), $S = 0.01$, and $S = 0.035$. Plugging into the estimate (11.1.4), we get

$$f(15,0) \approx 9.09 - (0.166)(-5) = 9.92, \quad f(15,0.01) \approx 9.09 - (0.166)(-5) - (52)(0.01) = 9.40,$$

$$f(15, 0.035) \approx 9.09 - (0.166)(-5) - (52)(0.035) = 8.10.$$

The real answers are 10.08 mg/L, 9.49 mg/L, and 8.14 mg/L respectively. These are reasonably good in view of how far away 15 is from 20. ■

Remark 11.1.6. If $f : \mathbf{R}^n \to \mathbf{R}$ has a local extremum at some $\mathbf{a}$ then we know that the partial derivatives $f_{x_j}(\mathbf{a})$ at $\mathbf{a}$ all vanish (Theorem 10.2.2), so for points $\mathbf{b}$ near $\mathbf{a}$ the partial derivatives $f_{x_j}(\mathbf{b})$ are all near $f_{x_j}(\mathbf{a}) = 0$. Hence, the linear approximation to f near such $\mathbf{b}$ takes the form

$$f(\mathbf{b}+\mathbf{h}) \approx f(\mathbf{b}) + ((\nabla f)(\mathbf{b})) \cdot \mathbf{h}$$

where the gradient $(\nabla f)(\mathbf{b})$ is *short* (its n entries $f_{x_j}(\mathbf{b})$ are all small, so its overall length is near 0).

If $c = f(\mathbf{b})$ labels the level set of f in which $\mathbf{b}$ lives then to make $f(\mathbf{b}+\mathbf{h})$ achieve a desired value nearer to the local extremum $f(\mathbf{a})$ – such as $c \pm 0.15$ (sign depending on whether $\mathbf{a}$ is a local maximum or local minimum) – we need the dot product $((\nabla f)(\mathbf{b})) \cdot \mathbf{h}$ to be equal to the signed increment step (such as ± 0.15). But how can we achieve this when $(\nabla f)(\mathbf{b})$ is very short? In general, how big in magnitude can a dot product $\mathbf{v} \cdot \mathbf{w}$ be for nonzero n-vectors $\mathbf{v}$ and $\mathbf{w}$? In terms of the angle θ between such vectors we have

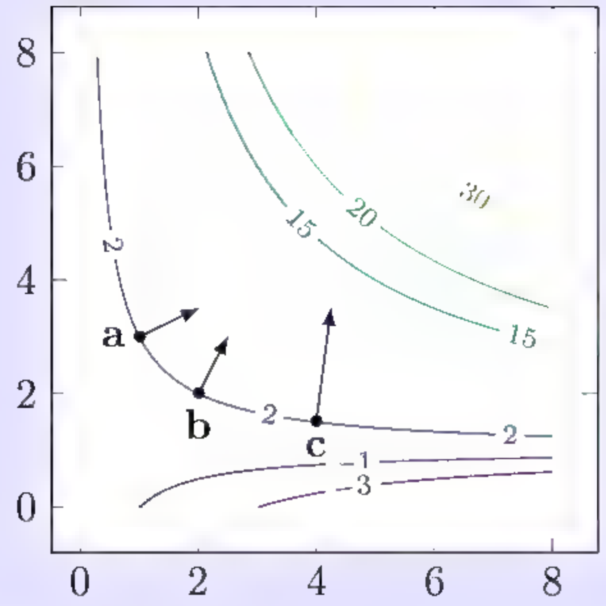
$$\mathbf{v} \cdot \mathbf{w} = \|\mathbf{v}\|\|\mathbf{w}\| \cos(\theta),$$

so the most positive (resp. negative) this can be is $\|\mathbf{v}\|\|\mathbf{w}\|$ (resp. $-\|\mathbf{v}\|\|\mathbf{w}\|$), achieved when $\theta = 0°$ (resp. $\theta = 180°$), as we saw in Example 2.2.3 Thus, as $(\nabla f)(\mathbf{b})$ becomes shorter (which occurs as we look at level sets ever closer to the local extremum $\mathbf{a}$) we need to make $\mathbf{h}$ *longer* (and to point more fully in the direction of the appropriately signed vector $\pm(\nabla f)(\mathbf{b})$) in order for the dot product to even have a chance of being equal to the desired step size (such as ± 0.15). This need for $\mathbf{h}$ to become longer is *exactly* accounting for observation (ii) in Example 10.2.13 that there is a "spreading out" among level sets with a fixed increment gap in the function values as we approach a local extremum.

11.2. The gradient as normal to contours. We now explain the visual significance of the gradient.

Theorem 11.2.1. Let $f : \mathbf{R}^2 \to \mathbf{R}$ be a scalar-valued function, and suppose $(\nabla f)(a, b) \neq \mathbf{0}$.

(i) The gradient $(\nabla f)(a, b)$ is *perpendicular to* the level set of f that goes through (a, b) (meaning: it is perpendicular to the tangent line of the level set through (a, b)). It points in the direction of maximal increase for $f(x, y)$ for (x, y) moving away from (a, b). The following figure demonstrates this with the function $f(x, y) = xy - x$.



A contour plot of $f(x, y) = xy - x$.
The gradient ∇f is drawn at three points:
$\mathbf{a} = (1, 3)$, $\mathbf{b} = (2, 2)$, $\mathbf{c} = (4, 3/2)$.
Observe that ∇f is perpendicular to the level curve.

(ii) The equation

$$(\nabla f)(a, b) \cdot \begin{bmatrix} x - a \\ y - b \end{bmatrix} = 0$$

in the (x, y)-plane is the line *tangent* to the curve in the contour plot of $f(x, y)$ through $(x, y) = (a, b)$. Explicitly, the equation of this line is

$$f_x(a, b)(x - a) + f_y(a, b)(y - b) = 0. \tag{11.2.1}$$

For an explanation of Theorem 11.2.1, see Section 11.4. A similar result (with a similar explanation) holds when $\mathbf{R}^2$ is replaced with $\mathbf{R}^3$:

Theorem 11.2.2. For a scalar-valued function $f : \mathbf{R}^3 \to \mathbf{R}$ and point $\mathbf{a}$ for which $(\nabla f)(\mathbf{a}) \neq \mathbf{0}$, the gradient vector is perpendicular to the *plane* tangent to the level set of f through $\mathbf{a}$. In particular, this tangent plane has the equation

$$(\nabla f)(a_1, a_2, a_3) \cdot \begin{bmatrix} x - a_1 \\ y - a_2 \\ z - a_3 \end{bmatrix} = 0. \tag{11.2.2}$$

As a special case, the graph of a function $h : \mathbf{R}^2 \to \mathbf{R}$ is the surface S with equation $z = h(x, y)$ that is the level set $f = 0$ of the function $f(x, y, z) = z - h(x, y)$ whose gradient $(-h_x, -h_y, 1)$ never vanishes (due to the third entry being 1). The tangent plane to S at a point $(a, b, h(a, b))$ then has equation $(-h_x(a, b), -h_y(a, b), 1) \cdot (x - a, y - b, z - h(a, b)) = 0$ that after some algebra becomes

$$z = h(a, b) + h_x(a, b)(x - a) + h_y(a, b)(y - b). \tag{11.2.3}$$

The result in (11.2.2) carries over to functions $f : \mathbf{R}^n \to \mathbf{R}$ for $n > 3$ as well, except that the concept of tangent plane to a level-set surface in $\mathbf{R}^3$ has to be replaced with an appropriate notion of "tangent hyperplane" to a level set $\{f = c\}$ in $\mathbf{R}^n$.

Example 11.2.3. Consider the circle defined by the equation $x^2 + y^2 = 25$. Let's find a normal vector to this circle at $(x, y) = (3, 4)$, as well as the equation of the tangent line to the circle at this point. This specific case can be done via single-variable calculus (try if you are interested), but we'll do it here using gradients as an illustration of a general method that applies to situations with 3 or more variables too.

Define $h(x, y) = x^2 + y^2$, so $h(3, 4) = 25$ and $\nabla h = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$. A normal vector to the level set $h = 25$ (this is our circle) through $(3, 4)$ is given by the gradient vector $(\nabla h)(3, 4) = \begin{bmatrix} 6 \\ 8 \end{bmatrix}$.

Now, if $\begin{bmatrix} x \\ y \end{bmatrix}$ is any point on the tangent line, the vector from $\begin{bmatrix} 3 \\ 4 \end{bmatrix}$ to $\begin{bmatrix} x \\ y \end{bmatrix}$ is *perpendicular* to the normal vector $\begin{bmatrix} 6 \\ 8 \end{bmatrix}$. As an equation, this says

$$\begin{bmatrix} x - 3 \\ y - 4 \end{bmatrix} \cdot \begin{bmatrix} 6 \\ 8 \end{bmatrix} = 0,$$

and so $6(x - 3) + 8(y - 4) = 0$ is the equation of the tangent line. After simplifying, this becomes $3x + 4y = 25$ as shown in Figure 11.2.1.

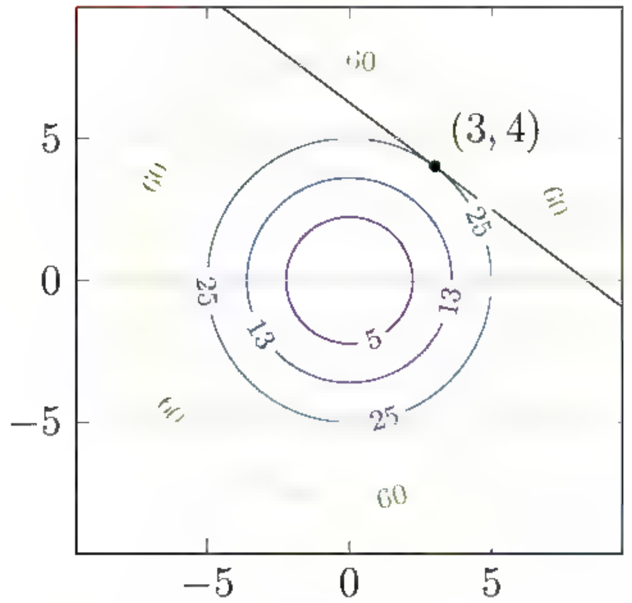


FIGURE 11.2.1. The level set of $h(x, y) = x^2 + y^2$ through $(3, 4)$ and its tangent line there.

■

Example 11.2.4. Consider the curve $x^2 + xy + y^3 = 7$. Let's find a nonzero vector perpendicular to the curve at the point $(2, 1)$ (this means: perpendicular to the tangent line of the curve at $(2, 1)$).

The curve is a level set of the function $h(x, y) = x^2 + xy + y^3$ shown in Figure 11.2.2, so the gradient vector $(\nabla h)(2, 1)$ is perpendicular to the curve at $(2, 1)$. Using partial derivatives of h we obtain that

$$(\nabla h)(x, y) = (2x + y, x + 3y^2),$$

so $(\nabla h)(2,1) = (5,5)$. Thus $(5,5)$ is such a vector.

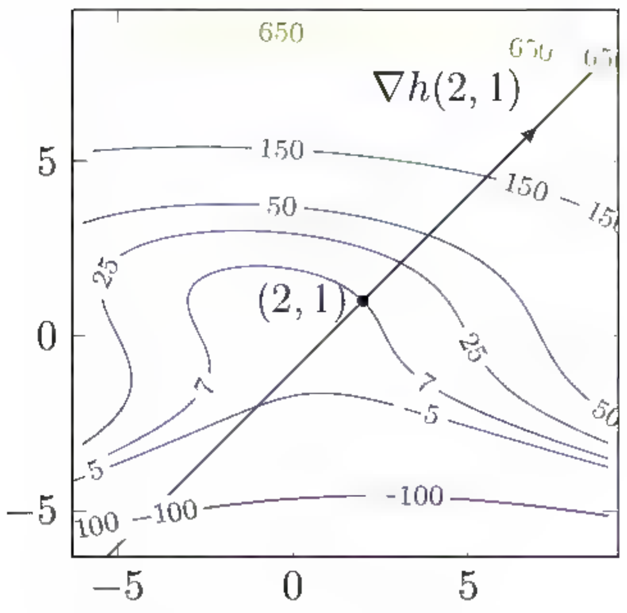


FIGURE 11.2.2. Contour plot for $h(x, y) = x^2 + xy + y^3$ including the level set through $(2, 1)$.

■

Example 11.2.5. Let's find a parametric representation for the line that intersects the curve $x^2+xy+y^3 = 7$ at the point $(2,1)$ perpendicularly to the curve there (meaning: perpendicular to the tangent line there).

Recall from (3.3.1) that the parametric equation of a line through a point $\mathbf{p}$ in the direction $\mathbf{v}$ is

$$\mathbf{x}(t) = \mathbf{p} + t\mathbf{v}.$$

In this problem $\mathbf{p} = (2, 1)$ is a point on the line. For the direction vector $\mathbf{v}$, we need a (nonzero) vector that is perpendicular to the tangent line of the curve at $(2, 1)$. We already found such a vector, namely $(5, 5)$, in Example 11.2.4. Thus,

$$\mathbf{x}(t) = \begin{bmatrix} 2 \\ 1 \end{bmatrix} + t \begin{bmatrix} 5 \\ 5 \end{bmatrix} = \begin{bmatrix} 2+5t \\ 1+5t \end{bmatrix}$$

is a parametric representation for the line. This line is shown in Figure 11.2.2. ■

Here is another example, now in three dimensions:

Example 11.2.6. Consider the sphere S given by the equation $x^2 + y^2 + z^2 = 6$. Let's find an equation for the tangent plane to S through the point $(2, 1, 1)$.

Let $f(x, y, z) = x^2 + y^2 + z^2$. Thus $\nabla f = \begin{bmatrix} 2x \\ 2y \\ 2z \end{bmatrix}$. As we saw in Theorem 11.2.2, this tangent plane is perpendicular to the gradient $(\nabla f)(2, 1, 1) = \begin{bmatrix} 4 \\ 2 \\ 2 \end{bmatrix}$ and so its equation is

$$\begin{bmatrix} 4 \\ 2 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} x-2 \\ y-1 \\ z-1 \end{bmatrix} = 0.$$

After simplifying, this becomes

$$2x + y + z = 6.$$

■

Example 11.2.7. Consider the surface S defined by $z = x^2 + y^2$. Let's find the equation of the tangent plane to S at the point $(1, 2, 5)$.

We can rewrite the equation as $x^2 + y^2 - z = 0$. Thus, the surface S is a level set of the function $f(x, y, z) = z - x^2 - y^2$. Hence, the gradient vector

$$(\nabla f)(x, y, z) = \begin{bmatrix} -2x \\ -2y \\ 1 \end{bmatrix}$$

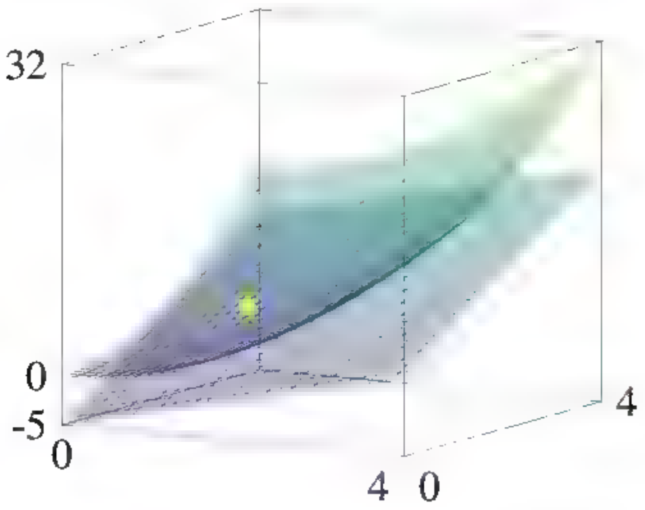
is perpendicular to the level set of f through each point (x, y, z). Thus, the gradient

$$(\nabla f)(1, 2, 5) = \begin{bmatrix} -2 \\ -4 \\ 1 \end{bmatrix}$$

is perpendicular to S at $(1, 2, 5)$. The tangent plane to the surface through $(1, 2, 5)$ is therefore perpendicular to $\begin{bmatrix} -2 \\ -4 \\ 1 \end{bmatrix}$, so its equation is obtained from the "point-normal" form:

$$\begin{bmatrix} -2 \\ -4 \\ 1 \end{bmatrix} \cdot \left(\begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \\ 5 \end{bmatrix} \right) = 0.$$

This simplifies to $-2(x - 1) - 4(y - 2) + (z - 5) = 0$, or equivalently $z = 2x + 4y - 5$.



Comparing the graph of $f(x, y) = x^2 + y^2$ and the tangent plane of f at (1,2,5). The tangent plane is given by $z = 2x + 4y - 5$.

FIGURE 11.2.3. A surface graph and the best linear approximation at the yellow point.

■

11.3. Gradient descent.

One of the main reasons to study calculus of n variables is to optimize multivariable functions; i.e., to find their local maxima and minima (with the eventual aim of finding global maxima and minima). Realistic problems of this type cannot be solved exactly; rather one needs ways to numerically approximate the answer. A powerful method for doing this is *gradient descent*; we give a brief introduction to it in this section. It will be discussed in the context of broader applications in Example 12.4.4 (to solve linear programming problems), Section G.3 (for learning in neural networks), and Appendix I (using Newton's method).

Disclaimer: realistic gradient descent requires a lot more care with approximation details and numerical methods than we have time to explain here. Our goal is to give a first exposure to some of the basic ideas, especially how knowledge of linear algebra illuminates one's understanding of the procedure.

Example 11.3.1. Imagine that a raindrop falls on a hill. It will head to the bottom – water always finds the lowest elevation. Or at the very least it will find a *local* minimum: it might not find the bottom of the hill, but it will find a lake half-way down the hill, perhaps. The raindrop certainly doesn't know anything about the geography of the hill. However, at every moment, it simply "chooses" to roll in the steepest possible direction. It is not only raindrops that find local minima. Physical systems find energy minima, and they do so by moving in the "direction" where potential energy decreases the most. ■

The preceding example suggests the following strategy, called *gradient descent*:

> to find the minimum of a function $f(x, y)$, do the following: move away from (x, y) in the direction in which f decreases the fastest.
>
> Similarly, to find the maximum of a function $f(x, y)$, do the following: move away from (x, y) in the direction in which f increases the fastest.

We need to make this mathematically precise. We can quantify *direction* (without regard to speed) by giving a unit vector $\mathbf{v} \in \mathbf{R}^2$; i.e., a vector of length $\|\mathbf{v}\| = 1$. To test "how fast f is increasing or decreasing in the direction of $\mathbf{v}$" we use the local approximation from (11.1.2) with $t \in \mathbf{R}$ near 0:

$$f(\mathbf{a} + t\mathbf{v}) \approx f(\mathbf{a}) + (\nabla f)(\mathbf{a}) \cdot (t\mathbf{v}) = f(\mathbf{a}) + t(\nabla f)(\mathbf{a}) \cdot \mathbf{v}.$$

This says that if we move a small distance $|t|$ in the direction $\mathbf{v}$ for $t > 0$ and in the direction of $-\mathbf{v}$ for $t < 0$ then the change in f is $\approx t\,((\nabla f)(\mathbf{a}) \cdot \mathbf{v})$, whose rate of change with respect to t (i.e., its t-derivative) is $(\nabla f)(\mathbf{a}) \cdot \mathbf{v}$. Thus, our question becomes:

> how do we choose a unit vector $\mathbf{v}$ so that $(\nabla f)(\mathbf{a}) \cdot \mathbf{v}$ is largest (and then work with $t > 0$ when seeking to maximize f and with $t < 0$ when seeking to minimize f)?

We can solve this by geometry! The dot product $(\nabla f)(\mathbf{a}) \cdot \mathbf{v}$ equals $\|(\nabla f)(\mathbf{a})\|\|\mathbf{v}\| \cos(\theta)$, where θ is the angle between $\mathbf{v}$ and $(\nabla f)(\mathbf{a})$. Clearly, the largest this gets is when $\cos(\theta) = 1$, and for that we want $\theta = 0°$. Likewise, the most negative it gets is when $\cos(\theta) = -1$, which says $\theta = 180°$. Therefore:

Theorem 11.3.2. Let $f : \mathbf{R}^n \to \mathbf{R}$ be a scalar-valued function and $\mathbf{a} \in \mathbf{R}^n$ a point at which the gradient $(\nabla f)(\mathbf{a}) \in \mathbf{R}^n$ is nonzero.

The gradient $(\nabla f)(\mathbf{a})$, or rather the associated unit vector $(\nabla f)(\mathbf{a})/\|(\nabla f)(\mathbf{a})\|$, is the direction in which f increases most rapidly at $\mathbf{a}$. Likewise, the opposite unit vector $-(\nabla f)(\mathbf{a})/\|(\nabla f)(\mathbf{a})\|$ is the direction in which f decreases most rapidly at $\mathbf{a}$.

Example 11.3.3. Many physical systems can be modeled using the gradient of a function. What this means is that an object of interest experiences a force that is given by

$$\text{force} = -(\nabla V)(x, y, z) \tag{11.3.1}$$

where $V(x, y, z)$ is called the "potential energy" function.

For example, the "potential energy" function for the gravitational force created by the sun has the form $V(x, y, z) = c/\sqrt{x^2 + y^2 + z^2}$ for a suitable constant $c > 0$ (We choose coordinates putting the sun at $(0, 0, 0)$.) In words, equation (11.3.1) says that *the object experiences a force along the direction where V is most rapidly decreasing.* That is to say, the object "wants" to go to where V is smallest. ■

Example 11.3.4. Numerical work on energy minimization problems (such as in molecular biology and computational chemistry, discussed in Example 10.1.2) and the maximum likelihood estimation method in statistics (for computing the "most likely' parameters in a probabilistic model) relies on gradient descent and its refinements.

> To reconstruct the image of a black hole from noisy data (noise due to atmospheric interference), the image restoration can be set up as the task of minimizing a certain "energy" [**BJZFDF1**, Sec. 4.2, (9)]. For this application to handle imaging for certain kinds of black holes, gradient descent is too slow and so a more refined optimization technique is needed; see Example 1.1.3. ■

Here are some worked numerical examples of gradient descent.

Example 11.3.5. Let us find a minimum of $f : \mathbf{R}^2 \to \mathbf{R}$

$$f(x, y) = x^2 - 3xy + 3y^2 + 5y + 2x.$$

First of all,

$$(\nabla f)(x, y) = \begin{bmatrix} 2x - 3y + 2 \\ -3x + 6y + 5 \end{bmatrix}.$$

We will start somewhere and move in the direction of the negative gradient; in other words, we will start off trying $\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{a}$, for some $\mathbf{a}$ that we hope is near the minimum, and then at each stage we will move from $\mathbf{a}$ to

$$\mathbf{a} + t(\nabla f)(\mathbf{a})$$

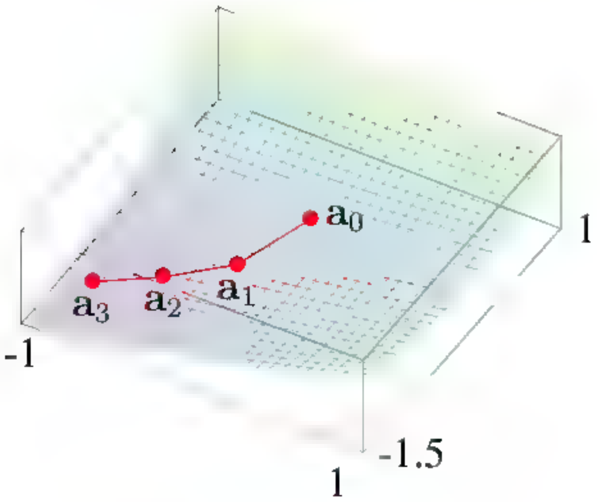
for some $t < 0$ with small absolute value. (The coefficient t is negative since we are seeking to decrease f rather than to increase f, as we are trying to minimize f.) In other words, we move in the direction of the negative gradient; the magnitude $|t|$ and the length $\|(\nabla f)(\mathbf{a})\|$ tell us how far we have moved.

We need to decide on two things:

(1) What is $\mathbf{a}$ – where do we start? We have no idea here; let's try $\mathbf{a} = (0, 0)$.
(2) What is t – how far do we go at each step? Again, we have no idea. In a realistic interpretation of gradient descent, one tries several values of t and seeks the best one. (In machine learning contexts, $|t|$ is sometimes called the *learning rate*.) In the bare-bones version here, let's use $t = -(0.1)$.

At each step, we will move from $\mathbf{a}$ to the point $\mathbf{a} - (0.1)(\nabla f)(\mathbf{a})$. Starting at $\mathbf{a} = (0, 0)$, we go through the following sequence of points in $\mathbf{R}^2$:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \rightsquigarrow \begin{bmatrix} -0.2 \\ -0.5 \end{bmatrix} \rightsquigarrow \begin{bmatrix} -0.51 \\ -0.76 \end{bmatrix} \rightsquigarrow \begin{bmatrix} -0.836 \\ -0.957 \end{bmatrix} \rightsquigarrow \cdots$$



The first three steps of gradient descent for f.
The starting point is $\mathbf{a}_0 = (0, 0)$.
The point $\mathbf{a}_n$ is determined by the rule:

$$\mathbf{a}_n = \mathbf{a}_{n-1} - (0.1)\nabla f(\mathbf{a}_{n-1}).$$

FIGURE 11.3.1. Initial steps of gradient descent may wander around

Nothing interesting is happening here: we seem to be just wandering around, as shown in Figure 11.3.1. But if we do it 10 times we get to the vector $\begin{bmatrix} -7.768 \\ -4.674 \end{bmatrix}$; if we do it 100 times we get $\begin{bmatrix} -8.835 \\ -5.245 \end{bmatrix}$, and if we do it 1000 times we get $\begin{bmatrix} -8.999\dots \\ -5.333 \end{bmatrix}$. It certainly looks like this is getting close to a specific point: $\mathbf{b} = \begin{bmatrix} -9 \\ -5\frac{1}{3} \end{bmatrix}$.

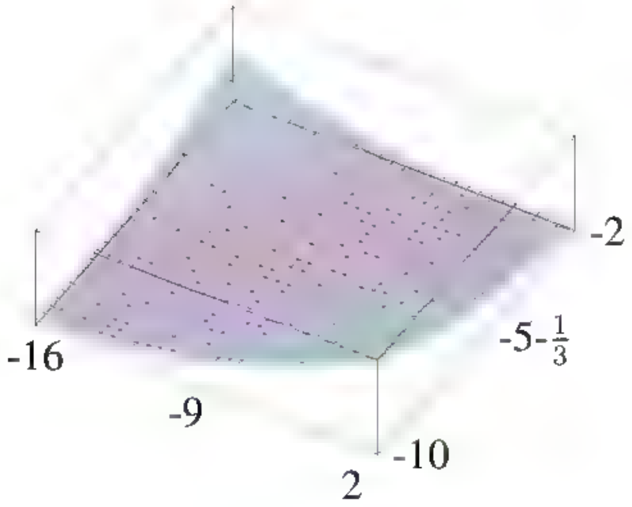
To verify this $\mathbf{b}$ is a special point of interest for the behavior of f, let us find the critical points symbolically. We must solve $\nabla f = \mathbf{0}$, or in other words

$$\begin{bmatrix} 2x - 3y + 2 \\ -3x + 6y + 5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The first equation tells us that $x = (3/2)y - 1$, and substituting this into the second equation gives

$$\frac{3}{2}y + 8 = 0,$$

so $y = -16/3 = -5\frac{1}{3}$ and hence $x = (3/2)y - 1 = -9$.



The point $(-9, -5\frac{1}{3})$
appears to be a minimum of f

FIGURE 11.3.2. A possible local minimum found via gradient descent

So far, this only says that $(-9, -5\frac{1}{3})$ is a critical point. You can convince yourself that it is a local minimum by checking a few values of (x, y) nearby, or by looking at Figure 11.3.2. In Chapter 26 we will do this more systematically (and give a genuine justification) via the "multivariable second derivative test"; see Example 26.4.2. ■

Example 11.3.6. Now let us try a somewhat more ambitious example, which would be much harder to solve by hand: let's find a local minimum of the function

$$f(x,y) = x + y - (0.1)(\ln(x) + \ln(y) + \ln(2x + 3y - 1) + \ln(3x + y - 1))$$

near $(x, y) = (1, 1)$. This might look like a big unmotivated mess, but actually it belongs to a class of functions that arise often in realistic optimization problems; we will come back to it from a broader perspective in Example 12.4.4 (with ε there equal to 0.1) and with another method in Example I.2.2.

The gradient of f is given by

$$\nabla f = \begin{bmatrix} 1 - (0.1)\left(1/x + 2/(2x + 3y - 1) + 3/(3x + y - 1)\right) \\ 1 - (0.1)\left(1/y + 3/(2x + 3y - 1) + 1/(3x + y - 1)\right) \end{bmatrix}.$$

Again, we will use the following algorithm:

for some negative t near 0, start at $\begin{bmatrix}1\\1\end{bmatrix}$ and at each stage replace $\mathbf{a}$ by $\mathbf{a}+t(\nabla f)(\mathbf{a})$.

We again choose $t=-(0.1)$. (In a real application one would use much care in choosing a good value of the "learning rate" $|t|$, possibly trying several different values and searching for the best one.)

So we start replacing $\mathbf{a}$ by $\mathbf{a}-(0.1)(\nabla f)(\mathbf{a})$ repeatedly. To three decimal digits, it looks like this:

$$\begin{bmatrix}1\\1\end{bmatrix} \rightsquigarrow \begin{bmatrix}0.925\\0.921\end{bmatrix} \rightsquigarrow \begin{bmatrix}0.852\\0.844\end{bmatrix} \rightsquigarrow \dots.$$

If you do this 10 times, you arrive at $\mathbf{b}\approx\begin{bmatrix}0.445\\0.370\end{bmatrix}$ with $(\nabla f)(\mathbf{b})\approx\begin{bmatrix}0.150\\0.287\end{bmatrix}$, which isn't all that much like $\begin{bmatrix}0\\0\end{bmatrix}$. If we do the same iteration 100 times, we arrive at $\mathbf{c}\approx\begin{bmatrix}0.455\\0.254\end{bmatrix}$, for which both coordinates of $(\nabla f)(\mathbf{c})$ vanish to at least three decimal places. (And if you do 1000 iterations, you arrive at a vector that agrees with $\mathbf{c}$ to six decimal places, so there's not much gain for the additional effort.)

In summary, $(0.455, 0.254)$ is (to good approximation) a critical point, which we found not by any symbolic computation but rather by the numerical method of repeatedly moving in the direction of the negative of the gradient.

What if we had started closer to the critical point, say at $\mathbf{a}=\begin{bmatrix}0.5\\0.3\end{bmatrix}$? In this case the process works a bit better: after 9 iterations it arrives at $\begin{bmatrix}0.4554573\dots\\0.2542429\dots\end{bmatrix}$, where ∇f vanishes to three decimal places, and another iteration yields $\begin{bmatrix}0.4554417\dots\\0.2542224\dots\end{bmatrix}$ that is essentially the same point (and at which ∇f is even a bit closer to $\mathbf{0}$). So with this new choice of $\mathbf{a}$, we reach the (approximate) critical point to three digits' accuracy after approximately 10 iterations rather than 100 iterations as above. ■

11.4. Explanation of properties of gradients. We now explain informally why the linear approximation (11.1.3) holds and why Theorem 11.2.1 holds. One can walk from (a,b) to (x,y) in two steps: first go along a path from (a,b) to (x,b) horizontally, and then go to (x,y) by moving vertically.

- In the first step we increase the x-coordinate from a to x holding y fixed, so

$$\text{change in } f \text{ in first step} \approx f_x(a,b)\times(x-a). \tag{11.4.1}$$

- Next, we move the y-coordinate from b to y holding x fixed, so

$$\text{change in } f \text{ in second step} \approx f_y(x,b)\times(y-b).$$

  Since (x,b) is very close to (a,b) (as $x\approx a$), we may expect that $f_y(x,b)$ and $f_y(a,b)$ are very close. Make the further approximation $f_y(x,b)\approx f_y(a,b)$, yielding

$$\text{change in } f \text{ in second step} \approx f_y(a,b)\times(y-b). \tag{11.4.2}$$

Now add (11.4.1) to (11.4.2) to get (11.1.3). By moving in more coordinate directions, this same reasoning explains the version for any number of variables in (11.1.2).

Next, let's turn our attention to explaining Theorem 11.2.1. Start at (a,b) and then move a little bit to a point (x,y) further along on the contour curve of f through (a,b). *By definition of the contour curve as a locus "$f(x,y)=c$" for some fixed value of c, such as $c=f(a,b)$*, f does not change at all: $f(x,y)=f(a,b)$.

Now recall the local approximation $f \approx f(a,b) + (\nabla f)(a,b) \cdot \begin{bmatrix} x-a \\ y-b \end{bmatrix}$ from (11.1.3). Applying this at the point (x,y) near (a,b) on the contour curve at which f has the value $f(a,b)$, we conclude that

$$(\nabla f)(a,b) \cdot \begin{bmatrix} x-a \\ y-b \end{bmatrix} = 0 \tag{11.4.3}$$

(i.e., the approximate deviation from $f(a,b)$ of the value f at (x,y) vanishes).

So *near* (a,b) the contour curve looks just like the line with equation (11.4.3). In other words, (11.4.3) must be the equation of the tangent line! But – as we just saw in Example 11.2.3 – (11.4.3) describes a line through (a,b) perpendicular to the vector $(\nabla f)(a,b)$. Therefore, the vector $(\nabla f)(a,b)$ is perpendicular to the contour curve of f through (a,b).

Chapter 11 highlights (links to highlights in and next chapters)

Notation	**Meaning**	**Location in text**
∇f	for $f : \mathbf{R}^n \to \mathbf{R}$, it is the vector-valued function $\mathbf{R}^n \to \mathbf{R}^n$ whose jth component function is $\partial f/\partial x_j$	Definition 11.1.1

Concept	**Meaning**	**Location in text**
gradient for $f : \mathbf{R}^n \to \mathbf{R}$ at $\mathbf{a} \in \mathbf{R}^n$	the n-vector whose jth entry is $(\partial f/\partial x_j)(\mathbf{a})$	Definition 11.1.1
linear approximation to $f : \mathbf{R}^n \to \mathbf{R}$ at $\mathbf{a} \in \mathbf{R}^n$	the approximation $f(\mathbf{a}) + ((\nabla f)(\mathbf{a})) \cdot (\mathbf{x} - \mathbf{a})$ to $f(\mathbf{x})$ for $\mathbf{x}$ near $\mathbf{a}$	(11.1.2), (11.1.3)
gradient descent	iterative process that moves in direction of negative gradient with the aim of reaching a (local) minimum	Section 11.3

Result	**Meaning**	**Location in text**
$(\nabla f)(\mathbf{a})$ is a normal vector to the level set of f through $\mathbf{a}$	for $f : \mathbf{R}^n \to \mathbf{R}$ with $n = 2, 3$, if $(\nabla f)(\mathbf{a}) \neq \mathbf{0}$ it is a normal vector to the tangent line/plane of the level curve/surface of f through $\mathbf{a}$	Thms 11.2.1, 11.2.2
directions of $\pm(\nabla f)(\mathbf{a})$ are directions of greatest increase and decrease for $f : \mathbf{R}^n \to \mathbf{R}$ at $\mathbf{a} \in \mathbf{R}^n$	if $(\nabla f)(\mathbf{a}) \neq \mathbf{0}$, unit vectors $\pm(\nabla f)(\mathbf{a})/\|(\nabla f)(\mathbf{a})\|$ point in the directions emanating from $\mathbf{a}$ in which f has the greatest increase (+) and decrease (−)	Theorem 11.3.2

Skill	**Location in text**
compute linear approximation to $f(\mathbf{a} + \mathbf{h})$ for small $\mathbf{h}$ using dot product against gradient vector $(\nabla f)(\mathbf{a})$	Examples 11.1.2, 11.1.3
compute equation of tangent line/plane to level set using (nonzero) gradient	Examples 11.2.3–11.2.7
carry out 1 or 2 steps of gradient descent in concrete situations (with simple numbers)	Example 11.3.5

12. Constrained optimization via Lagrange multipliers

The problem of *finding the largest or smallest value* of a function occurs across many fields. You have seen many examples of this with single-variable functions. However, modeling real-world problems often requires functions involving several variables; e.g., "minimize the function $x^2 - 3x + y^2 + 4y + 2$ over all $(x, y) \in \mathbf{R}^2$." A new feature in the multivariable case is that it is often necessary to maximize or minimize *subject to a constraint*; e.g., "minimize $x^3 + 2xy^2$ subject to the constraint $3x^2 + 4y^2 = 1$."

Examples of this problem arise in many situations, such as:

- in economics, to maximize profits subjects to a budgetary constraint (or maximize expected investment return subject to a given initial wealth to be spread across a portfolio);
- in natural sciences, to maximize the outcome of a physical process (distance, heat produced, etc.) subject to a given amount of energy available;
- in drug design, to maximize binding affinity for the molecular structure subject to constraints expressing desired chemical properties (e.g., [**KP**, Sec. 5.4] has such examples in $\mathbf{R}^{256}$!).

A basic 2-variable version of the problem (which moreover contains many of the essential mathematical issues of the general multivariable case) is this:

find the maximum of $f(x, y)$ *subject to the constraint* $g(x, y) = c$ (for a specific $c \in \mathbf{R}$)

where g is some auxiliary function of prior interest (i.e., the constraint $g(x, y) = c$ expresses some real-world condition on the points (x, y) of interest for the optimization problem). Geometrically, this means that we are trying to maximize f but restricting (x, y) to a certain curve – a level curve of g. For example, it could be: "find the extrema of $x^2 - y$ subject to the constraint $x^3 + x + y^3 + y = 1$."

Alternatively, we might want to work with a constraint of the form

find the maximum of $f(x, y)$ *subject to the constraint* $g(x, y) \leq c$ (for a specific $c \in \mathbf{R}$)

An example of why we might want to do this is in Example 12.4.2. Such inequality constraints also show up in machine learning algorithms to make classifiers, in work with support vector machines.

One approach is to try to solve for y in terms of x under the constraint: that is, we start with $g(x, y) = c$, try to solve for y in terms of x, substitute it into $f(x, y)$, and apply single-variable calculus to $f(x, y(x))$. This doesn't work well when g is complicated (it is often impossible to explicitly solve for y in terms of x on the constraint curve $g(x, y) = c$, and even when possible typically $f(x, y(x))$ is a mess). A better method, developed in this chapter, is called *Lagrange*[13] *multipliers*.

By the end of this chapter, you should be able to:

- set up an equation for a constrained optimization problem in any number of variables by using Lagrange multipliers;
- solve the "Lagrange multiplier" equations in examples in 2 or 3 variables.

12.1. Motivation. Here are two important examples of optimization subject to a constraint.

Example 12.1.1 (Maximizing utility). Suppose a consumer with an amount w ("wealth") to spend wants to buy some of each of n commodities. If p_i is the price (say in dollars) of 1 unit of the ith commodity and x_i is the amount of the ith commodity that is purchased then the total amount spent by the consumer is $p_1x_1 + p_2x_2 + \cdots + p_nx_n$. In economics, one measures the value of such a purchasing decision in terms of a "utility function" $U(x_1, \dots, x_n)$ whose meaning and mathematical form can vary depending on the circumstances. The goal is always to maximize $U(x_1, \dots, x_n)$ subject to wealth or budgetary constraints. The constraint is typically $\sum_{i-1}^{n} p_ix_i \le w$ (can't spend more than one's wealth), or $\sum_{i-1}^{n} p_ix_i = w$. (The latter can handle the general case: one commodity could be "cash reserves" for the money not spent on these goods.)

---

[13] Joseph-Louis Lagrange (1736-1813) was an Italian-French mathematician who made fundamental contributions in both mathematical physics and number theory. The Euler–Lagrange equations in the calculus of variations and Lagrangian mechanics (a powerful reformulation of Newtonian mechanics avoiding preferred coordinates) are named in his honor.

The *Cobb–Douglas model* (due to the mathematician Charles Cobb and economist/politician Paul Douglas in the first half of the 20th century, and used for comparing production costs against output value for companies) takes U to have the form $Ax_1^{a_1}x_2^{a_2}\cdots x_n^{a_n}$ for a coefficient $A > 0$ and fractional exponents $a_1, \dots, a_n > 0$ (typically determined through a least-squares fitting process against data). We'll treat a situation with a specific U in Example 12.2.7 after we have introduced the necessary mathematical techniques. ■

Example 12.1.2. In numerous machine learning applications (as varied as computational biology, image classification, and text classification), a key problem is for a machine to "learn" how to classify n-vector data points (e.g., genomic sequences, images, text, etc.) into one of two types (e.g. cancerous or not, face or not, political ad or not). This can be formulated as a constrained optimization problem in n variables, with n typically very large. The algorithms that solve it are referred to by the awful name "support vector machines" (SVM). SVM's involve multiple constraints at once, which requires more linear algebra, so we'll come back to it in Example 19.4.4. ■

In single-variable calculus, we solve for *local* extrema of a function $f(x)$ on an interval $[a, b]$ by searching among the values $x = c$ for which $f'(c) = 0$, along with the endpoints of the interval (where f' may not vanish, but which could nonetheless be a local extremum for f on $[a, b]$). In the multivariable setting, the method we introduce below will likewise identify two types of candidates for local extrema *under a constraint*: one type satisfying an analogue of the equation $f' = 0$ in the single-variable case, and one type satisfying an analogue of the "endpoint" condition in the single-variable case.

As a warm-up, we first consider an example for which we can easily find the solutions to a constrained optimization problem by inspection yet can also see an essential geometric property of such solutions that illustrates the method to be used in the general case.

Example 12.1.3. Consider the unit sphere S centered at the origin $\mathbf{0}$ in $\mathbf{R}^3$, and $f(x, y, z) = z$. The sphere S is defined by $x^2 + y^2 + z^2 = 1$, so for $g(x, y, z) = x^2 + y^2 + z^2$ finding the extrema for z *on the sphere* S is the same as finding the points in the region $g = 1$ at which f attains maximal or minimal values.

By staring at the sphere S, we see by inspection that f attains its maximal value on S at the north pole $\mathbf{p} = (0, 0, 1)$ and attains its minimal value on S at the south pole $-\mathbf{p} = (0, 0, -1)$. We emphasize that these two points on S are *not* local extrema for f on the ambient $\mathbf{R}^3$: for tiny $t > 0$, at points $(0, 0, 1 + t)$ close to $\mathbf{p}$ the value $1 + t$ of f there is bigger than the maximal value $f(\mathbf{p}) = z(\mathbf{p}) = 1$ for $f = z$ on S, and similarly at points $(0, 0, -1 - t)$ close to $-\mathbf{p}$ the value $-1 - t$ of f there is less than the minimal value $f(-\mathbf{p}) = z(-\mathbf{p}) = -1$ on S. Such points $(0, 0, \pm(1 + t))$ are *not in* S and so are *irrelevant* for the purpose of optimizing f subject to the constraint $g = 1$ that defines S.

We have just observed a very important point, so make sure you understand it: a solution to a constrained optimization problem on $\mathbf{R}^n$ of the form "optimize $F(\mathbf{x})$ subject to the condition $G(\mathbf{x}) = c$" is typically *not* a local extremum for F on the ambient $\mathbf{R}^n$. Hence, there is *no reason* for it to be a critical point of F; i.e., no reason for ∇F to vanish there! Indeed, in our situation above with the sphere S and $f(x, y, z) = z$, the gradient $(\nabla f)(\mathbf{x})$ is equal to the *nonzero* vector $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ for all $\mathbf{x} \in \mathbf{R}^3$.

But there is nevertheless something special about the behavior of ∇f at the points $\pm\mathbf{p}$ where f has its extreme values on the region S defined by $g = 1$. To explain it, we will work with the gradient ∇g at points of the constraint region S. For any point $\mathbf{x} = (x, y, z) \in \mathbf{R}^3$ (which may or may not lie in S) we have $(\nabla g)(\mathbf{x}) = \begin{bmatrix} 2x \\ 2y \\ 2z \end{bmatrix} = 2\mathbf{x}$. In Figure 12.1.1 we show $S = \{\mathbf{x} \in \mathbf{R}^3 : g(\mathbf{x}) = 1\}$ with the corresponding gradient vector $(\nabla g)(\mathbf{x}) = 2\mathbf{x}$ drawn in at every point $\mathbf{x} \in S$: this is a vector perpendicular to the tangent plane of S at $\mathbf{x}$, pointing outward from the sphere at $\mathbf{x}$ with length 2 (the lengths don't all look the same in the Figure 12.1.1 due to the effect of perspective). The non-negative coordinate axes are also drawn, in light blue, but that is irrelevant in what follows.

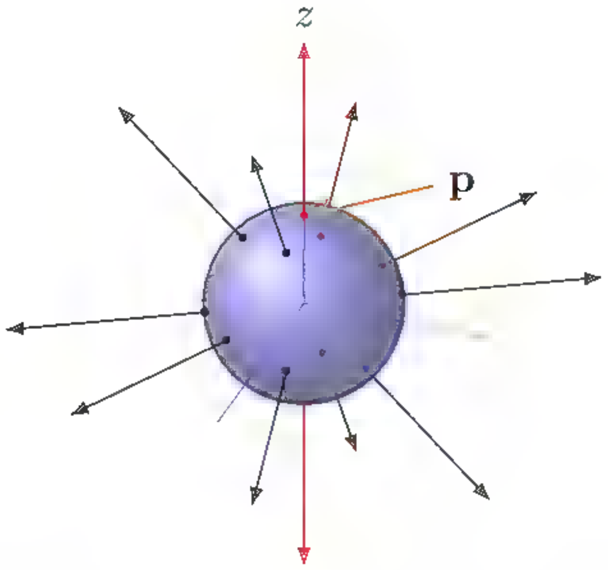


FIGURE 12.1.1. A sphere S with radial vectors of length 2 emanating from every point.

Here is the key observation: for each $\mathbf{x} \in S$, compare the "radial" line through $\mathbf{x}$ along the direction of $(\nabla g)(\mathbf{x})$ and the *vertical* line through $\mathbf{x}$ along the direction of $(\nabla f)(\mathbf{x}) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$. The points $\mathbf{a} \in S$ for which these two lines through $\mathbf{a}$ coincide are exactly the north pole $\mathbf{p}$ and south pole $-\mathbf{p}$ (indicated by the red radial vectors in Figure 12.1.1)! The equality of such lines associated with some point $\mathbf{a} \in S$ – equality of the line spanned by $(\nabla f)(\mathbf{a})$ and the line spanned by $(\nabla g)(\mathbf{a})$ – says exactly that $(\nabla f)(\mathbf{a})$ is a scalar multiple of the nonzero gradient vector $(\nabla g)(\mathbf{a})$ (with scalar multiplier equal to 1/2 in this situation, but the specific scalar isn't essential). This special property at the constrained extrema, that the vector $(\nabla f)(\mathbf{a})$ is a scalar multiple of the vector $(\nabla g)(\mathbf{a})$, will turn out to be a general feature of solutions to *all* constrained extremum problems, as Theorem 12.2.1 below makes precise. Such a relation among gradient vectors will play the role for constrained optimization that the first derivative test does in single-variable calculus (and that the vanishing of the gradient does for unconstrained optimization on $\mathbf{R}^n$). ■

Next we consider an example in which we can't compute the constrained extrema by inspection as with the sphere in Example 12.1.3, but for which we can visualize a relationship among gradients at the constrained extrema similar to what we observed in Example 12.1.3 at the poles of the unit sphere.

Example 12.1.4. For $g(x, y) = x^4 + x^3y + y^2$, let's find the *local* extrema of $f(x, y) = x^3 + xy^2$ subject to the constraint $g(x, y) = 1$; this is studying the behavior of $f(x, y)$ on the curve C defined by $g(x, y) = 1$ that is shown in Figure 12.1.2. By "local" extrema of $f(x, y)$ on C we mean points P on C for which the points $(x, y) \in C$ near P all satisfy $f(P) \geq f(x, y)$ (constrained local maximum at P) or all satisfy $f(P) \leq f(x, y)$ (constrained local minimum at P).

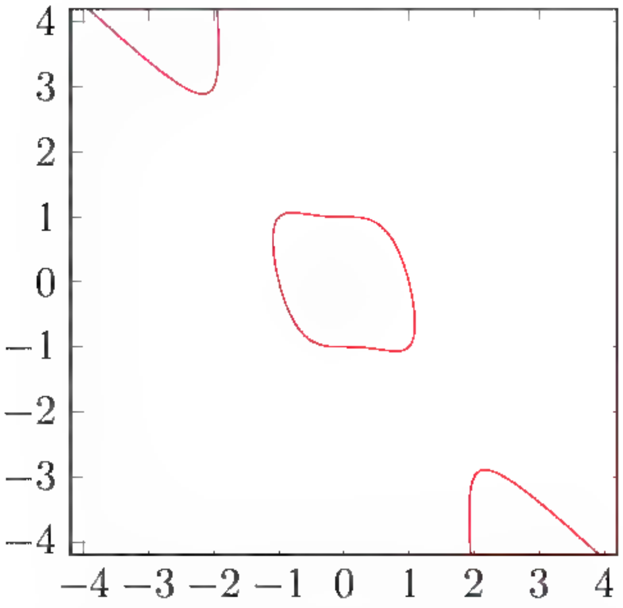


FIGURE 12.1.2. Constraint curve C defined by $x^4 + x^3y + y^2 = 1$.

To visualize the task of finding local extrema for $f(x, y)$ on C, we shall consider the level curves $f(x, y) = m$ and allow m to *vary*. For some values of m the curve $f(x, y) = m$ does not touch C, so f never attains the value m on C. For other values of m, the curve $f(x, y) = m$ touches C; the points where this happens are where f has value m on C.

The utility of this is illustrated in Figure 12.1.3 which shows a blue level curve $f(x, y) = m_0$ for a specific $m_0 \approx 2.056$ and black level curves $f(x, y) = m$ for m near m_0. (The contour labels show that the value of f changes rapidly!) For m near m_0, if $m > m_0$ then the curve $f(x, y) = m$ does not touch C whereas if $m < m_0$ then the curve $f(x, y) = m$ touches C at a couple of points and finally the curve $f(x, y) = m_0$ touches C at exactly one point P_0 (black dot), moreover in a "tangential" manner (i.e., the tangent lines at P_0 to the blue curve $f(x, y) = m_0$ and the red curve C coincide). To summarize:

(i) for $Q \in C$ near P_0 we have $f(Q) \leq m_0 = f(P_0)$, so $f(x, y)$ on C has a local maximum at P_0;
(ii) the curve $f(x, y) = m_0$ through the constrained local extremum P_0 is "tangent" to C at P_0;
(iii) the curve $f(x, y) = m$ for nearby $m < m_0$ is *not* tangent to C at the points Q near P_0 where this level curve of f meets C.

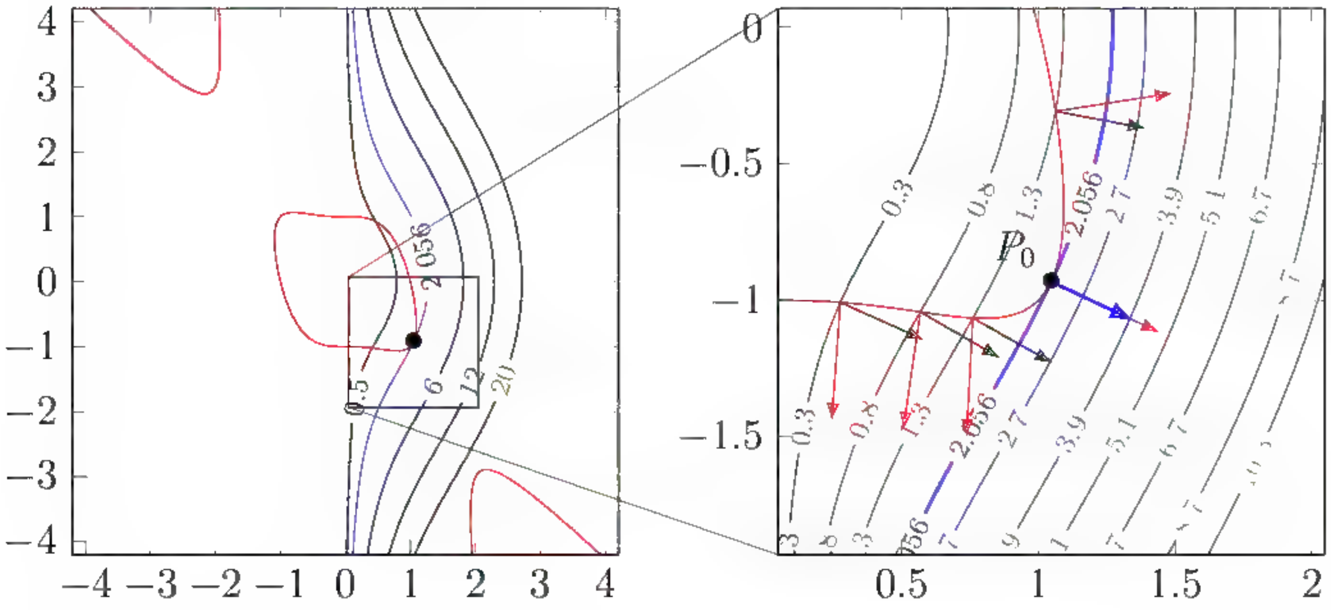


FIGURE 12.1.3. Red constraint curve C and level curves $f(x, y) = m$ for m near a specific $m_0 \approx 2.056$. The blue curve $f(x, y) = m_0$ is *tangent* to C at a point P_0 (black dot on the left) that is a constrained local maximum for f on C. At points of C near P_0 on level curves of f, compare ∇g (red) and ∇f (black or blue) on the right: they're aligned only at P_0!

The key is to reinterpret (ii) and (iii) in terms of gradient vectors. For the tangent line to a level curve at a point, the perpendicular direction is that of the gradient vector at that point (Theorem 11.2.1!). The equality of two lines through a point in $\mathbf{R}^2$ (such as tangent lines to curves at a common point, as above) is the same as equality of their perpendicular directions through the point, so we can restate (ii) and (iii):

(ii′) $(\nabla f)(P_0)$ and $(\nabla g)(P_0)$ span the same line;
(iii′) for points $Q \in C$ near P_0 but distinct from P_0, the vectors $(\nabla f)(Q)$ and $(\nabla g)(Q)$ do *not* point along the same line.

Hence, for points $(a, b) \in C$ near P_0, the condition

$$(\nabla f)(a, b) \text{ is a scalar multiple of } (\nabla g)(a, b)$$

(reminiscent of Example 12.1.3 at the north and south poles) holds at P_0 and *not* anywhere else nearby on C! We haven't said what the scalar multiplier is, but the fact that such a relationship holds between the two gradients at a point of C (regardless of the value of the scalar multiplier!) is a *very restrictive condition* that picks out P_0 from all other nearby points of C. This will lead to an analogue for constrained local extrema of the first derivative test in single-variable calculus (a criterion for finding *candidates* for local extrema; such candidates can fail to be local extrema when a second derivative vanishes). We will revisit this constrained optimization problem in Example 12.2.13. ■

12.2. The method of Lagrange multipliers.

Theorem 12.2.1. Suppose $f : \mathbf{R}^n \to \mathbf{R}$ and $g : \mathbf{R}^n \to \mathbf{R}$ are functions, and consider the problem of finding a local maximum (or local minimum) of f on the region where $g(\mathbf{x}) = c$. If a local extremum for f on the constraint region (i.e., the region of $\mathbf{x}$ satisfying $g(\mathbf{x}) = c$) occurs at $\mathbf{a}$ then either

$$(\nabla g)(\mathbf{a}) = \mathbf{0} \text{ or } (\nabla f)(\mathbf{a}) = \lambda\,(\nabla g)(\mathbf{a}) \tag{12.2.1}$$

for some scalar λ (called the "Lagrange multiplier") that may depend on $\mathbf{a}$.

Warning: We do not know λ; we (often) need to solve for it! The equations $\nabla f = \lambda \nabla g$ in $\mathbf{R}^n$ and $g(\mathbf{x}) = c$ in $\mathbf{R}$ amount to $n + 1$ scalar equations in $n + 1$ variables $x_1, \ldots, x_n, \lambda$, and we often need to solve for them all. (Sometimes we're lucky and λ cancels out in the calculations, so we don't need to find λ.) In real-world problems, λ can have a useful interpretation; see Remark 12.2.14.

This theorem requires practice to digest, since it involves an auxiliary quantity λ that isn't part of the initial problem we have been trying to solve. The existence of λ expresses a substantial *geometric* condition at constrained local extrema $\mathbf{a}$ (which we observed in Figure 12.1.3 at P_0): the vector $(\nabla f)(\mathbf{a})$ lies in the line spanned by $(\nabla g)(\mathbf{a})$ when this latter gradient vector is nonzero. We are going to use this "multiplier equation" (or really both options in (12.2.1)) as a technique to solve constrained extrema problems. This should be regarded as an analogue to solving extrema problems in single-variable calculus by using the vanishing condition in the first derivative test (and solving unconstrained extrema problems in multivariable calculus by using the vanishing of the gradient; i.e., testing critical points).

To summarize: any solution to an optimization problem must satisfy some *auxiliary equation* (vanishing of gradient in the unconstrained case, one of the options in (12.2.1) in the constrained setting), and we will solve this equation to arrive at *candidates* for solutions to the original optimization problem. We can compare values at those candidate points to see what are the biggest and smallest, just as we do with the first derivative test in single-variable calculus. A wrinkle with the multiplier equation in (12.2.1) is that it involves the existence of an additional unknown scalar λ, so "solving" (12.2.1) is a new kind of problem. The many worked examples below show that there are systematic ways to handle this.

It is a pleasant fact of experience, as we will see in the examples below, that usually $(\nabla g)(\mathbf{x})$ is non-vanishing on the *entire* constraint region defined by $g(\mathbf{x}) = c$. In such happy situations the case

$(\nabla g)(\mathbf{a}) = \mathbf{0}$ in (12.2.1) is impossible, so we can then ignore it and focus on the multiplier equation $(\nabla f)(\mathbf{a}) = \lambda(\nabla g)(\mathbf{a})$. Complementing Example 12.1.4, here is another visualization of the multiplier equation:

Example 12.2.2. Let $f(x,y) = 2x^3y - xy^5$, and let $g(x,y) = x^2 + y^2$. For $c > 0$, the constraint curve $g(x,y) = c$ is $x^2 + y^2 = c$, which we recognize as the circle of radius $\sqrt{c}$ centered at $\mathbf{0}$. Note that $\nabla g = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$ is non-vanishing on $g = c$, so by Theorem 12.2.1 any local extremum $\mathbf{a}$ for f along $g = c$ must satisfy the multiplier equation $(\nabla f)(\mathbf{a}) = \lambda(\nabla g)(\mathbf{a})$ for some scalar λ (typically depending on $\mathbf{a}$). With a bit of algebra one finds that $(\nabla f)(x,y) = \begin{bmatrix} 6x^2y - y^5 \\ 2x^3 - 5xy^4 \end{bmatrix}$ is *nonzero* for $(x,y) \neq \mathbf{0}$, so the relation $(\nabla f)(\mathbf{a}) = \lambda(\nabla g)(\mathbf{a})$ among *nonzero* gradients says that they point along the *same* (or *opposite*) directions (depending on whether λ is positive or negative). This visual interpretation is insensitive to the specific value of λ, which is good since we don't know (yet) what λ means. We push the visualization further via Figure 12.2.1 that shows (in green/yellow, red, and blue) the contour plot of f for positive values of f, with the level curves $f = b$ for $b > 0$ having one part in each quadrant, and superimposed on a contour plot for g whose level curves are the circles centered at $\mathbf{0}$ (just some of which are shown).

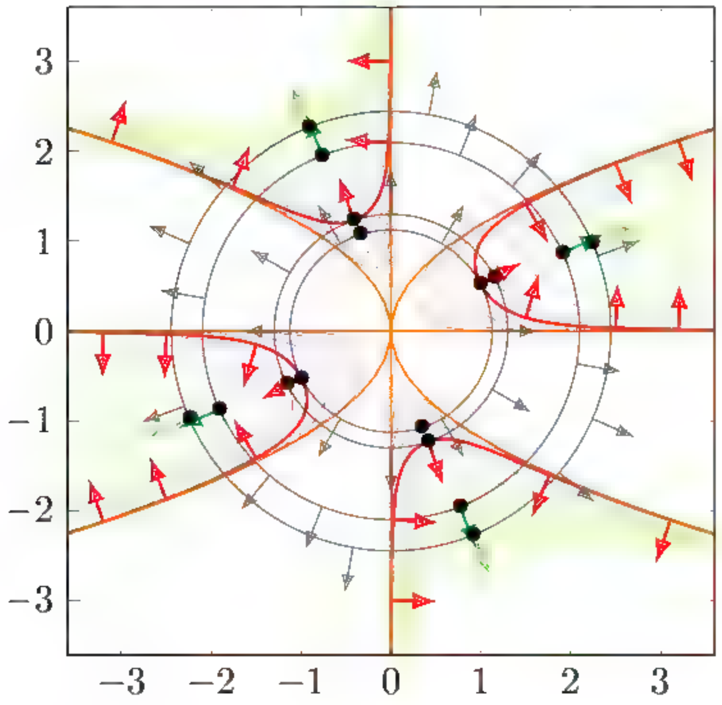


FIGURE 12.2.1. Contour plot of $f = 2x^3y - xy^5$ showing the multi-part curves $f = b$ for $b \geq 0$: orange for $b = 0$, blue for $0 < b < 1$, red for $b = 1$, green/yellow for $b > 1$. The curve $f = 0$ consists of $x = 0$, $y = 0$, and horizontal parabolas $x = \pm y^2/\sqrt{2}$, dividing the plane into 8 regions with f positive on 4 of them as shown. As $b > 0$ grows, the curve $f = b$ moves "outward"; black dots are local maxima of f on a constraint curve $x^2+y^2 = c$.

On the level curves of g (the circles) we have drawn a small vector in the direction of ∇g at many points, and likewise with ∇f's on the red level curve $f = 1$ and at black dots on some green level curves $f = b$ with $b > 1$ (we draw small vectors since direction is all that matters for interpreting when $(\nabla f)(\mathbf{a})$ and $(\nabla g)(\mathbf{a})$ point along the same line). As $b > 0$ grows, the curve $f = b$ moves outward, corresponding to ∇f's pointing to the "outward" side of the curve $f = b$ (relative to $\mathbf{0}$).

On each circle $g = c$, there are marked points (black dots). These are where f attains a local maximum *along the circle* $g = c$. If you look closely, you'll see such points are exactly where the short vectors along the level curves of f and g through a common point are aligned (pointing along the same direction, in fact)! This is most visible on the red level curve $f = 1$ for which we have drawn in the direction of many of the gradient vectors ∇f along the curve, though it is also visible at the black dots on some other level curves $f = b$ since as b increases the level curves $f = b$ are "moving outwards" (ensuring that on a given circle $g = c$, the "last" b for which the circle touches a level curve $f = b$ is a local maximum for f on the circle, and ∇f there is in the direction of ∇g there).

If you think of $b > 0$ as corresponding to "time" and $f = b$ as a shape of a curve at time b, then as b grows we can imagine the curves $f = b$ evolving outwards. For increasing time b, look at where and how each part of $f = b$ crosses a chosen circle $g = c$. The "last time" that part of the contour plot of f touches a chosen circle $g = c$ is a local maximum on that circle, and the touching is tangential there (meaning that the curves have the same tangent line at such a point). This tangential property corresponds to the Lagrange multiplier equation for local extrema of f on $g = c$, for reasons we encountered in Example 12.1.4 and will discuss in Remark 12.2.5 (it amounts to using Theorem 11.2.1).

What about the local *minima* for f on a chosen circle $g = c$? In the spaces "between" the 4 shown parts of the contour plot of f for positive f-values in Figure 12.2.1 is where the level curves $f = b$ for $b < 0$ would appear if we had drawn them. For $b < 0$ increasing towards 0, those (not drawn) parts of the contour plot are "moving inwards" toward $\mathbf{0}$. Thinking about $b < 0$ as "time in the past", in any of the 4 parts of the plot where $f < 0$ the "first time" a chosen circle $g = c$ is touched by a level curve for f is a (negative) local minimum for f on that circle. Alignment of gradients (and tangential behavior) happens at these constrained local minima, just as we saw above at the constrained local maxima, completing our visualization of the multiplier equation at constrained local extrema. ■

Remark 12.2.3. As with the first derivative test, (12.2.1) identifies *candidates* for constrained **local** extrema. It can happen there are no constrained **global** extrema! For instance, the function xy on the parabola $y = x^2 - 3$ has a local minimum at $(1, -2)$ but no global minimum: plugging $y = x^2 - 3$ into the function xy yields $x^3 - 3x$ that we can analyze via single-variable calculus to see that it has a local minimum at $x = 1$, but $x^3 - 3x$ has arbitrarily negative values as $x \to -\infty$ and so there is no global minimum. For applications in economics, physics, biology, etc., we usually have non-mathematical reasons to know there is a maximum or minimum on the constraint region, so we won't dwell on this issue (but it is important for a complete mathematical treatment, which involves additional ideas).

In Examples 12.1.4 and 12.2.2 we gave visual evidence for why Theorem 12.2.1 holds, and a more precise argument is given in Section 12.5. There is a generalization of Theorem 12.2.1 for r constraints $g_1(\mathbf{x}) = c_1, \dots, g_r(\mathbf{x}) = c_r$ with any $r \geq 1$; this arises in economics and many scientific (and mathematical) problems. The result for more than one constraint requires a concept that has not yet been introduced ("linear independence"), so we will come back to it later (in Section 19.4).

We now illustrate Theorem 12.2.1 in numerical examples. The first one below can also be done by single-variable calculus: try that way and compare the answers via both methods. The subsequent examples are unpleasant or hopeless to solve using single-variable calculus.

Example 12.2.4. Let's find the points on the line $x + 2y = 20$ that are extrema of xy (as a function on this line). Here, $f(x, y) = xy$, $g(x, y) = x + 2y$, and $c = 20$.

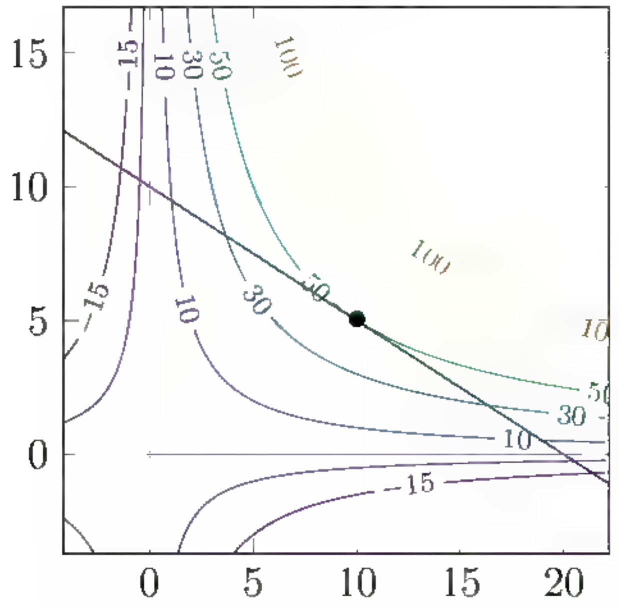
We compute that $(\nabla f)(x, y) = \begin{bmatrix} y \\ x \end{bmatrix}$, and the gradient $(\nabla g)(x, y)$ is equal to the vector $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ everywhere, so ∇g never vanishes. Thus, the first possibility in (12.2.1) cannot happen, so (12.2.1) becomes just the Lagrange multiplier condition

$$\begin{bmatrix} y \\ x \end{bmatrix} = \lambda \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

for some λ we don't yet know. Thus, $y = \lambda$ and $x = 2\lambda$. But what are the possibilities for λ? Now *remember we have the constraint*

$$g(x, y) = x + 2y = 20.$$

So we plug "$(x, y) = (2\lambda, \lambda)$" into that to get $4\lambda = 20$, so $\lambda = 5$ and hence $x = 10, y = 5$. Therefore, the only possible extremal value is at $(10, 5)$; the value $f(10, 5) = 50$ is in fact a maximum on $g = 20$.



The line $x + 2y = 20$ and
a contour plot of $f(x, y) = xy$.
Observe that the line is tangent to the $f = 50$ contour.

FIGURE 12.2.2. An extremum for $f(x, y)$ under a linear constraint is at a point where a level set for f is tangent to that linear constraint.

■

Remark 12.2.5. Let's see why it is no coincidence that at the constrained extremum $\mathbf{a} = (10, 5)$ for xy under the linear constraint in Figure 12.2.2, the constraint line $g = 20$ is *tangent* to the level set for xy through that point; i.e., the line is tangent to the hyperbola $xy = f(x, y) = f(\mathbf{a}) = 50$ passing through $\mathbf{a}$.

More broadly, consider the general setting of the theorem on Lagrange multipliers with $n = 2$ (with any f and g, so in particular not just g in the special form $c_1x + c_2y$ as above). If ∇f and ∇g are both non-vanishing on the constraint region (as often happens) – so the first option in (12.2.1) does not occur – then we claim that for any local extremum $\mathbf{a}$ of f along $g(x, y) = c$, the level curve $f(x, y) = f(\mathbf{a})$ for f through $\mathbf{a}$ and the level curve $g = c$ through $\mathbf{a}$ *touch tangentially* at the common point $\mathbf{a}$. This means (by definition of "touch tangentially") that the tangent lines at $\mathbf{a}$ to the level curves for f and g through $\mathbf{a}$ coincide. In concrete terms, it says that as we visualize the level curves $f = b$ for *varying* b, we seek those that touch the constraint curve $g = c$ tangentially, with such points $\mathbf{a}$ of tangential contact being the points produced by the method of Lagrange multipliers. This is the tangential observation that we made in Examples 12.1.4 and 12.2.2 and saw illustrated in Figures 12.1.3 and 12.2.1.

The key to understanding such tangency at constrained extrema is the fact from Theorem 11.2.1 that nonzero gradient vectors for f and g at any point $\mathbf{b}$ are perpendicular to the tangent line to the respective level sets for f and g passing through $\mathbf{b}$. Equivalently, the tangent line directions at $\mathbf{b}$ on the level sets $f = f(\mathbf{b})$ and $g = g(\mathbf{b})$ are perpendicular to $(\nabla f)(\mathbf{b})$ and $(\nabla g)(\mathbf{b})$ respectively. Why is this relevant?

The reason is that, as we discussed and visualized in Examples 12.1.4 and 12.2.2, at a constrained extremum $\mathbf{a}$ these nonzero gradient vectors point along the *same or opposite direction* (by the theorem on Lagrange multipliers!), so they span the *same* line through $\mathbf{0}$ and hence the directions perpendicular to these gradients *coincide*. This common direction orthogonal to the gradients is parallel to the tangent lines at $\mathbf{a}$ on each of the level curves $f(x, y) = f(\mathbf{a})$ and $g(x, y) = g(\mathbf{a}) = c$ through $\mathbf{a}$ (again by Theorem 11.2.1). Since these two tangent lines are each parallel to the same direction, they are parallel to each other. These tangent lines both pass through the point $\mathbf{a}$, and if two parallel lines share a common point then they must be the same line! Hence, the tangent lines at $\mathbf{a}$ on the level curves coincide, as we claimed.

Remark 12.2.6. The tangential perspective on solving Lagrange multiplier problems visually in the 2-variable case as in Remark 12.2.5 comes up in economics in two different ways: one with linear g and non-linear f (as in Example 12.2.4), and another with linear f and non-linear g (Example 12.2.12 will give a 3-variable analogue, phrased in a geometric context).

In consumer theory, one maximizes a non-linear utility function f subject to a linear constraint $g = c$ that encodes a fixed budget. The level sets of f are then called "indifference curves", so solving the Lagrange multiplier problem visually amounts to looking at the indifference curves $f = b$ for various b and seeking those that touch the line $g = c$ at a point of tangency to that line. This is exactly what happens in Figure 12.2.2 (where $b = 50$ is the one magic number).

In producer theory, a linear cost function f is minimized subject to a given level of output that is a non-linear function g of variables x and y corresponding to labor and capital inputs. The level set for $g = c$ is called an "isoquant" (presumably for "*same quant*ity of output"), and the level sets $f = b$ for varying b constitute a collection of *parallel lines*. The corresponding visual approach to the Lagrange multiplier method is to look among those parallel lines to find any that cross the given isoquant $g = c$ tangentially. The points of tangential contact are the points a that are sought.

Example 12.2.7. Here is an instance of Example 12.1.1, but phrased in mathematical form to avoid assuming familiarity with specialized economics terminology. Consider the function

$$u(x_1, x_2) = x_1^{1/3} x_2^{2/3}$$

for $x_1, x_2 > 0$. For a fixed $w > 0$ ("wealth") and $p_1, p_2 > 0$ ("prices"), we want to maximize u subject to the constraint $p_1x_1 + p_2x_2 = w$. One might imagine this with explicit numbers in the roles of p_1, p_2, w, but it is *much better* for us to work out the answer as a general formula in terms of p_1, p_2, w since (i) this makes the algebraic work cleaner, (ii) this is the form of such answers that is actually used in practice.

Letting $g(x_1, x_2) = p_1x_1 + p_2x_2$, the constraint is $g(x_1, x_2) = w$. Note that at every (x_1, x_2) the gradient of g is the same vector $\begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$ that is nonzero (just as in Example 12.2.4). The gradient of u is

$$\nabla u = \begin{bmatrix} (1/3)x_1^{-2/3} x_2^{2/3} \\ (2/3)x_1^{1/3} x_2^{\ 1/3} \end{bmatrix},$$

and at a point (x_1, x_2) where a maximum occurs on the constraint curve $g = w$, we have $\nabla u = \lambda \nabla g$ for some scalar λ. This says

$$\begin{bmatrix} (1/3)x_1^{\ 2/3} x_2^{2/3} \\ (2/3)x_1^{1/3} x_2^{\ 1/3} \end{bmatrix} = \lambda \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} \lambda p_1 \\ \lambda p_2 \end{bmatrix}.$$

Equating corresponding vector entries and remembering the constraint equation, we arrive at 3 equations in 3 unknowns:

$$\frac{1}{3} x_1^{-2/3} x_2^{2/3} = \lambda p_1, \ \ \frac{2}{3} x_1^{1/3} x_2^{-1/3} = \lambda p_2, \ \ p_1x_1 + p_2x_2 = w. \tag{12.2.2}$$

This is probably not a type of system of equations you have worked with before, but because the exponents $2/3$ and $1/3$ in $u(x_1, x_2) = x_1^{1/3} x_2^{2/3}$ sum to 1 it will turn out that the algebra to be done is not too bad.

A useful rule of thumb is to try to "solve for λ" using as many expressions as possible, and then by equating all of those to each other we may make some progress. In both the first and second equations in (12.2.2) we "solve for λ" to get the chain of equalities

$$\frac{1}{3p_1} x_1^{-2/3} x_2^{2/3} = \lambda = \frac{2}{3p_2} x_1^{1/3} x_2^{-1/3}.$$

Equating the first and third expressions and moving denominators and factors with negative exponents over to the other side yields

$$p_2x_2 = 2p_1x_1.$$

Thus, the third equation in (12.2.2) becomes

$$w = p_1x_1 + p_2x_2 = 3p_1x_1,$$

so $x_1 = w/(3p_1)$. Likewise, $w = p_1x_1 + p_2x_2 = (1/2)p_2x_2 + p_2x_2 = (3/2)p_2x_2$, so $x_2 = 2w/(3p_2)$. Hence, the extremum (which happens to be a maximum, as desired) occurs at

$$\left(\frac{w}{3p_1}, \frac{2w}{3p_2}\right). \tag{12.2.3}$$

■

Remark 12.2.8. In Example 12.2.7, it was very convenient for algebraic purposes that the exponents $1/3$ and $2/3$ add up to 1. The general Cobb–Douglas economic model for two production costs x_1 and x_2 (of which Example 12.2.7 is a very specific instance) involves maximizing a function of the form $U(x_1, x_2) = Ax_1^{a_1}x_2^{a_2}$ for some $A > 0$ and some exponents $a_1, a_2 > 0$ with $a_1 + a_2$ generally *not* equal to 1. But there is a trick: since raising to a positive exponent preserves the direction of inequalities among positive numbers, maximizing U is the same as maximizing $U^{1/(a_1+a_2)} = A^{1/(a_1+a_2)}x_1^{a_1/(a_1+a_2)}x_2^{a_2/(a_1+a_2)}$. The gain is that this latter function has exponents $a_1/(a_1 + a_2)$ and $a_2/(a_1 + a_2)$ that *do* sum to 1! The general Cobb–Douglas model is discussed at length in Econ 50.

Example 12.2.9. Let's find the point(s) on the curve $\{(x, y) \in \mathbf{R}^2 : 8y^2 - 4x^3 + x^4 = 0\}$ closest to the point $P = (3, 0)$, and compute that minimal distance. The quantity to minimize is $\sqrt{(x-3)^2 + y^2}$, the distance between (x, y) and $P = (3, 0)$. It is the same as minimizing the quantity inside the square root, so we focus on minimizing $f(x, y) = (x - 3)^2 + y^2$ subject to the constraint $g(x, y) = 0$ for $g(x, y) = 8y^2 - 4x^3 + x^4$.

By the Lagrange multiplier method, any point (x, y) at which f attains a local extremum on $g = 0$ (either local maximum or local minimum, though we are seeking the global minimum) either makes ∇g vanish or ∇f a scalar multiple of ∇g.

Let's first figure out where (if anywhere!) ∇g vanishes on the constraint curve $g = 0$, so such "bad" points (corresponding to the first option in (12.2.1)) can be treated separately. We calculate

$$\nabla g = \begin{bmatrix} -12x^2 + 4x^3 \\ 16y \end{bmatrix},$$

so for this to equal $\mathbf{0}$ says $-12x^2 + 4x^3 = 0$ and $16y = 0$, so $y = 0$ and $0 = -12x^2 + 4x^3 = 4x^2(-3 + x)$. This happens at the points $(0, 0)$ and $(3, 0) = P$. But P isn't on the constraint curve $g = 0$ (indeed, $g(P) = g(3, 0) = -4 \cdot 27 + 81 = -108 + 81 = -27 \neq 0$), so really only $(0, 0)$ occurs in this case (and $g(0, 0) = 0$).

At any other point on the constraint curve we must be in the second option of (12.2.1):

$$\nabla f(x, y) = \lambda \nabla g(x, y) \tag{12.2.4}$$

for some unknown scalar λ. We have computed the gradient of g above, and $\nabla f = \begin{bmatrix} 2(x-3) \\ 2y \end{bmatrix}$, so the Lagrange multiplier condition (12.2.4) becomes

$$\begin{bmatrix} 2(x-3) \\ 2y \end{bmatrix} = \lambda \begin{bmatrix} -12x^2 + 4x^3 \\ 16y \end{bmatrix} = \lambda \begin{bmatrix} 4x^2(-3+x) \\ 16y \end{bmatrix}.$$

Equating corresponding vector entries and remembering the constraint equation, we want to find solutions to the combined system

$$2x-6=\lambda(4x^2(-3+x)),\ \ 2y=\lambda(16y),\ \ 8y^2-4x^3+x^4=0. \tag{12.2.5}$$

At this point we carry out a very useful *general technique*: use each of the conditions other than the constraint equation to obtain different expressions for λ which we then equate to get new conditions on the variables without involving λ. An extremely important point that one must **always** keep in mind to be systematic about this step is that one must always *be careful about division by zero* (i.e., avoid it!). To be more specific, the first and second equations in our combined system give expressions

$$\frac{2x-6}{4x^2(x-3)}=\lambda=\frac{2y}{16y}=\frac{1}{8} \tag{12.2.6}$$

assuming the denominators $4x^2(x-3)$ and $16y$ are both nonzero. The left side is $1/(2x^2)$ upon cancelling $x-3$ provided that $x\neq 3$, as is necessary for the original fraction makes sense.

It is always good to first determine when one of those denominators vanishes. (In some such cases there might not be a λ satisfying all conditions in (12.2.5), but don't worry about it.) We handle that now:

Case 1. The case $4x^2(x-3)=0$ turns the first equation in our combined system (12.2.5) into $2x-6=0$, which is to say $x=3$. Then the constraint $g(x,y)=0$ becomes $g(3,y)=0$, which says $8y^2-27=0$. Hence, these problematic points are $(3,\pm\sqrt{27/8})$.

Case 2. The case $16y=0$ or equivalently $y=0$ makes the second equation in (12.2.5) tell us nothing, but the constraint $g(x,y)=0$ says $g(x,0)=0$, or in other words $-4x^3+x^4=0$. Since $-4x^3+x^4=x^3(x-4)$, this makes $x=0$ or $x=4$, yielding $(0,0)$ and $(4,0)$.

To summarize, so far we have obtained the points $(0,0)$, $(4,0)$, and $(3,\pm\sqrt{27/8})$ that merit separate treatment, and otherwise we have the two expressions $1/(2x^2)$ and $1/8$ for λ as obtained in (12.2.6) (assuming the non-vanishing of the denominators). Equating these two fractional expressions $1/(2x^2)$ and $1/8$ for λ gives $x^2=4$, which is to say $x=\pm 2$. The constraint curve $g(x,y)=0$ says $g(2,y)=0$ when $x=2$ and $g(-2,y)=0$ when $x=-2$. Since $g(2,y)=8y^2-4(2)^3+2^4=8y^2-16$ and $g(-2,y)=8y^2-4(-2)^3+(-2)^4=8y^2+48$, the latter never vanishes (so the case $x=-2$ doesn't occur!) and the former vanishes when $y^2=2$, which is to say $y=\pm\sqrt{2}$. Hence, we obtain the additional candidate points $(2,\pm\sqrt{2})$ for local extrema of f on the constraint curve.

Putting it all together, we have six points to examine: $(2,\pm\sqrt{2})$, $(3,\pm\sqrt{27/8})$, $(0,0)$, $(4,0)$.

Evaluating the function f at these points, we get $f(2,\pm\sqrt{2})=(2-3)^2+(\pm\sqrt{2})^2=1+2=3$, $f(3,\pm\sqrt{27/8})=0^2+(\pm\sqrt{27/8})^2=27/8$, $f(0,0)=9$, and $f(4,0)=1$. The smallest of these values is 1 attained at $(4,0)$, and the biggest is 9 attained at $(0,0)$. So the closest point to $P=(3,0)$ on the constraint curve $g=0$ is $(4,0)$ with distance $\sqrt{1}=1$, and the farthest point to P on the constraint curve is $(0,0)$ with distance $\sqrt{9}=3$. ■

Remark 12.2.10. In the preceding example, although we were seeking point(s) on a curve $g(x,y)=0$ closest to a given point P not on the curve, the method also yielded a unique point on the curve at a largest distance from P. The notable feature is that this point at maximal distance was the point $(0,0)$ which emerged in our analysis in the situation $\nabla g=\mathbf{0}$, and at this point ∇f is equal to $\begin{bmatrix}-6\\0\end{bmatrix}\neq\mathbf{0}$, so there is *no* scalar λ making $(\nabla f)(0,0)=\lambda(\nabla g)(0,0)$!

This illustrates the first case in (12.2.1) really can occur at a solution to a constrained optimization problem (such as a point at maximal distance): in the Lagrange multiplier theorem, the solution might be completely missed by the multiplier condition "$\nabla f=\lambda(\nabla g)$" and only captured by the other possibility

$\nabla g = \mathbf{0}$ in (12.2.1). Hence, the first option in (12.2.1) really must never be disregarded; it could be the only part of the method that actually finds the solution!

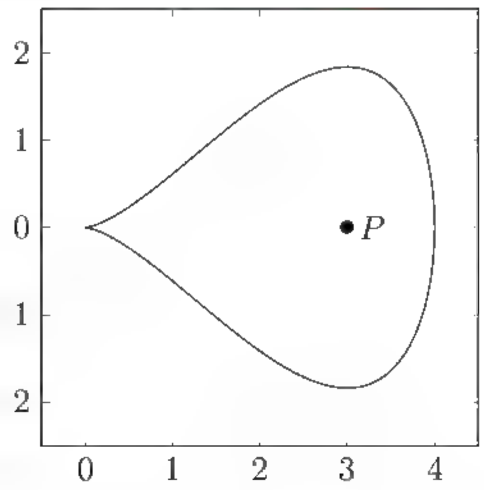
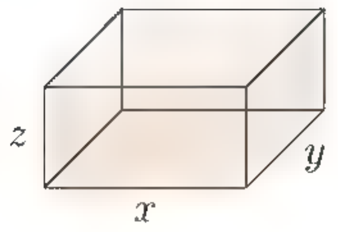


FIGURE 12.2.3. The curve $g(x,y) = 8y^2 - 4x^3 + x^4 = 0$ and the point $P = (3,0)$.

Figure 12.2.3 shows what the constraint curve $g = 0$ in Example 12.2.9 looks like, and this makes rather visible that $(4,0)$ is the point on this curve nearest to $(3,0)$, and that $(0,0)$ is the point on this curve farthest from $(3,0)$ (the pinching of the curve $g = 0$ at $(0,0)$ is related to the fact that ∇g vanishes there). But please don't conclude that working out Example 12.2.9 was a waste of time, on the grounds that we could have solved it by "looking at a picture". The point of this and the other worked examples is to give you *practice with a general technique*. We work out examples in 2-variable (and 3-variable) settings to give experience, but the importance of the method is that it is systematic and applicable to constrained optimization problems in *any* number of unknowns (where one generally can't draw any picture).

Many constrained optimization problems in economics, natural sciences, and computer science involve far more than 3 unknowns. Moreover, even in situations with 2 unknowns it might be impractical or impossible to solve such a problem by looking at a picture: the region might be too large, or there may be *auxiliary parameters* in a problem and we need a solution in terms of those parameters for applications to a wide array of parameter values (and hence it is impossible to proceed by drawing a picture, since a picture only treats specific parameter values and we can't look at infinitely many pictures).

Example 12.2.11. Suppose we want to make an open-topped rectangular cardboard box with volume 4, using as little material as possible. In other words, among rectangular boxes with volume 4, for what side lengths is the total area of the sides and the bottom (ignoring the top!) as small as possible?



A box with no lid.

Denote the side-lengths of the (rectangular) base of the box by x and y and the height of the box by z. The area of the base is xy. The front and back each have area xz, and the other two sides each have area yz. So the total area is

$$A(x,y,z) = xy + 2xz + 2yz$$

(we don't have $2xy$ since the top face is missing: it is an open box). The volume of the box is $V(x,y,z) = xyz$ that we are constraining to equal 4. We want to minimize $A(x,y,z)$ subject to this constraint (that $xyz = 4$).

We calculate that

$$\nabla A = \begin{bmatrix} y+2z \\ x+2z \\ 2x+2y \end{bmatrix}, \quad \nabla V = \begin{bmatrix} yz \\ xz \\ xy \end{bmatrix}.$$

By the theorem on Lagrange multipliers, at a minimum either $\nabla V = \mathbf{0}$ or $\nabla A = \lambda \nabla V$ for some λ. Since $xyz = 4$, none of x, y, or z can be zero. Consequently (from the expression for ∇V), we see that $\nabla V \neq \mathbf{0}$ on the entire constraint region. So $\nabla A = \lambda \nabla V$ for some unknown scalar λ:

$$\begin{bmatrix} y+2z \\ x+2z \\ 2x+2y \end{bmatrix} = \lambda \begin{bmatrix} yz \\ xz \\ xy \end{bmatrix}.$$

In other words, for some number λ necessarily the following must all hold:

$$y + 2z = \lambda yz, \quad x + 2z = \lambda xz, \quad 2x + 2y = \lambda xy. \tag{12.2.7}$$

Now it is again time to use the same handy technique as in Example 12.2.9: we "solve for λ" in each of the scalar equations in (12.2.7) arising from the Lagrange multiplier condition, and then equate those expressions to each other (to obtain new relations among x, y, z without reference to λ). We emphasize once more that in all such calculations, one must be vigilant about accounting for situations where we might divide by 0. Namely, solving each of the equations in (12.2.7) for λ gives

$$\lambda = \frac{y+2z}{yz}, \quad \lambda = \frac{x+2z}{xz}, \quad \lambda = \frac{2x+2y}{xy}$$

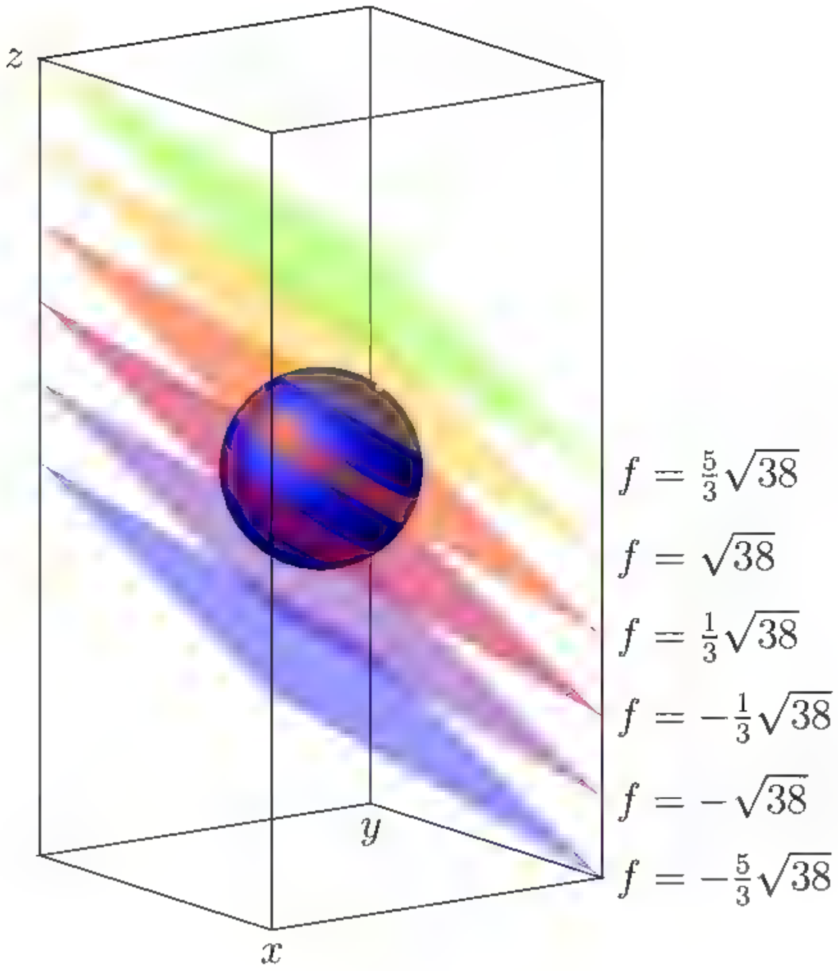
provided of course that the denominators are all nonzero. In our situation we know from the constraint equation $xyz = 4$ that each of x, y, z is nonzero, so indeed the denominators are all nonzero. Hence, the three expressions for λ are all legitimate, and equating them all to each other gives a triple equality

$$\frac{y+2z}{yz} = \frac{x+2z}{xz} = \frac{2x+2y}{xy}.$$

In the first equality, cancelling z from the denominator on both sides leaves us with $1 + 2(z/y) = 1 + 2(z/x)$, so after a tiny bit of algebra we get $x = y$. Similarly, the second equality yields (after cancelling x from the denominator on both sides) that $x/z + 2 = 2(x/y) + 2$, which after a tiny bit of algebra yields $y = 2z$. Thus $x = y = 2z$,

Having whittled down the three variables x, y, z to all be expressed in terms of one of them (namely, z), we get control on the remaining free variable by *remembering the constraint*: plugging all of this into the constraint equation gives $4 = xyz = (2z)(2z)z = 4z^3$, so $z = 1$ and thus $x = y = 2$. In other words, the open box of volume 4 and minimal surface area has height 1 unit and length and width of 2 units each. (We have actually shown just that $(x, y, z) = (2, 2, 1)$ is the only *candidate* for a constrained minimum. If we believe that a constrained minimum really exists – as seems "physically plausible" – then it must be this unique candidate (2,2,1). A complete mathematical justification that (2,2,1) is the constrained minimum involves more work.) ■

Example 12.2.12. Let's find the largest and smallest values taken by $2x + 3y + 5z$, when (x, y, z) belongs to the sphere $x^2 + y^2 + z^2 = 1$. Here $f(x, y, z) = 2x + 3y + 5z$ and $g(x, y, z) = x^2 + y^2 + z^2$.



z
y
x
$f = \frac{5}{3}\sqrt{38}$
$f = \sqrt{38}$
$f = \frac{1}{3}\sqrt{38}$
$f = -\frac{1}{3}\sqrt{38}$
$f = -\sqrt{38}$
$f = -\frac{5}{3}\sqrt{38}$

FIGURE 12.2.4. Six level sets for a linear function f; the yellow and purple ones ($f = \pm\sqrt{38}$) tangent to the blue sphere $g = 1$ touch it at points found by Lagrange multipliers.

The gradient ∇g is equal to $\begin{bmatrix} 2x \\ 2y \\ 2z \end{bmatrix}$ that only vanishes at the origin, which is not a point on the constraint region $g = 1$. So the first option in (12.2.1) cannot occur, and the Lagrange multiplier condition says $\begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \lambda \begin{bmatrix} 2x \\ 2y \\ 2z \end{bmatrix}$ for some unknown scalar λ. In particular, necessarily $\lambda \neq 0$ (since the left side is nonzero), so it makes sense to divide both sides by 2λ; this shows that $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$ is a multiple of $\begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$ (with the multiplier $1/(2\lambda)$), or in other words $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$ lies on the span ℓ of $\begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$.

But since (x, y, z) lies on the unit sphere, the vector $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$ has unit length, and on any line through the origin there are exactly two unit vectors (pointing in opposite directions). Hence, up to a sign $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$ is obtained from $\begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$ by dividing by its length: $\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \pm\frac{1}{\sqrt{4+9+25}} \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \pm\frac{1}{\sqrt{38}} \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$. By evaluating

f at these two points and checking which is larger, we see that $(1/\sqrt{38})\begin{bmatrix}2\\3\\5\end{bmatrix}$ corresponds to the maximum (where $f = \sqrt{38}$) and $-1/(\sqrt{38})\begin{bmatrix}2\\3\\5\end{bmatrix}$ corresponds to the minimum (where $f = -\sqrt{38}$). The level sets for f through these constrained extrema are tangent to the constraint sphere $g = 1$ at those points, as shown in Figure 12.2.4, illustrating a higher-dimensional version of the tangency in Remark 12.2.5.

We could have done this problem in another way: f is the dot product $\begin{bmatrix}2\\3\\5\end{bmatrix}\cdot\begin{bmatrix}x\\y\\z\end{bmatrix}$, and the dot product of two vectors is the product of their lengths *times* the cosine of the angle between them. So if we fix the length of $\begin{bmatrix}x\\y\\z\end{bmatrix}$ to be 1, the dot product is maximized when the angle between them is 0; i.e., $\begin{bmatrix}x\\y\\z\end{bmatrix} = a\begin{bmatrix}2\\3\\5\end{bmatrix}$ for some scalar a, as we found above (using the unit-vector condition on (x, y, z) to determine a up to a sign). ■

Example 12.2.13. In real-world constrained optimization, one can't solve the Lagrange multiplier equations explicitly as we have been able to do in the preceding examples (and as will happen in most homework and all exams in this course). Instead, software is used to find (approximate) solutions. The same is true of realistic unconstrained optimization even in single-variable calculus: the examples in calculus courses are specially designed to make the algebra work out nicely (to give you hands-on experience).

For a more realistic scenario where one doesn't solve the Lagrange multiplier equation by hand, let's revisit the constrained optimization problem from Example 12.1.4: for $g(x, y) = x^4 + x^3y + y^2$, we seek local extrema for $f(x, y) = x^3 + xy^2$ subject to the constraint $g(x, y) = 1$. First, let's check that ∇g is non-vanishing at all points of the curve $g = 1$. Since $(\nabla g)(x, y) = \begin{bmatrix}4x^3 + 3x^2y\\x^3 + 2y\end{bmatrix} = \begin{bmatrix}x^2(4x + 3y)\\x^3 + 2y\end{bmatrix}$, a bit of algebra shows that this vanishes only at $(0, 0)$, $(\pm\sqrt{8/3}, \mp(4/3)\sqrt{8/3})$, none of which lie on $g = 1$. Hence, at any local extremum $\mathbf{a}$ of $f(x, y)$ subject to $g(x, y) = 1$ we have $(\nabla f)(\mathbf{a}) = \lambda(\nabla g)(\mathbf{a})$ for some (unknown) scalar λ (which may depend on $\mathbf{a}$). Writing $\mathbf{a} = (a, b)$, this says

$$\begin{bmatrix}3a^2 + b^2\\2ab\end{bmatrix} = \lambda\begin{bmatrix}4a^3 + 3a^2b\\a^3 + 2b\end{bmatrix},$$

so for the 3 unknowns a, b, λ we have the simultaneous scalar equations

$$3a^2 + b^2 = \lambda(4a^3 + 3a^2b),\ \ 2ab = \lambda(a^3 + 2b),\ \ a^4 + a^3b + b^2 = 1 \tag{12.2.8}$$

(the final equation is the constraint equation, which we must never forget!).

As usual, we can rewrite the first two as the combined conditions

$$\frac{3a^2 + b^2}{4a^3 + 3a^2b} = \lambda = \frac{2ab}{a^3 + 2b}$$

provided that the denominators are both nonzero. Let's show such vanishing can't occur. If the first denominator were to vanish then the first equation in (12.2.8) would force $3a^2 + b^2 = 0$, so $a = 0$ and $b = 0$, but the constraint equation at the end of (12.2.8) rules out the possibility $(a, b) = (0, 0)$. If the second denominator were to vanish then the second equation in (12.2.8) would force $2ab = 0$, so either $a = 0$ or $b = 0$. But the assumed vanishing of the denominator $a^3 + 2b$ along with the vanishing of at least one of a or b would then force the vanishing of *both* a and b, which we have already noted is ruled out by the constraint $g(a, b) = 1$.

So the two fractional expressions for λ are legitimate, and by equating them and cross-multiplying we obtain (after a bit of algebra) $3a^5 - 8a^4b - 5a^3b^2 + 6a^2b + 2b^3 = 0$. Together with the constraint equation $g(a, b) = 1$, we conclude that (a, b) satisfies both conditions

$$3x^5 - 8x^4y - 5x^3y^2 + 6x^2y + 2y^3 = 0, \quad x^4 + x^3y + y^2 = 1. \tag{12.2.9}$$

Using the Wolfram command NSolve ("numerically solve"), there are 4 solutions (a, b) to the simultaneous equations in (12.2.9) and approximations to them are in the table below that also gives the quadrant in which each lies; the bottom row gives the approximate value of f at each such point.

a	1.04679	2.00793	−1.04679	−2.00793
b	−0.931593	−2.98514	0.931593	2.98514
Quadrant	IV	IV	II	II
$f(a, b)$	2.056	25.988	−2.056	−25.988

As a visualization, in Figure 12.2.5 we draw in various colors the level curves $f(x, y) = m$ through each of these points found by the Lagrange multiplier method. The tangency of the red constraint curve $g(x, y) = 1$ and the level curve of f through the point is quite visible in all cases. The values of f increase from very negative in the upper left to very positive in the lower right, so for the level curve $f = m_0$ through the tangency point on the yellow curve ($m_0 \approx -25.988$) or green curve ($m_0 \approx 2.056$) the values m near m_0 for which the curve $f = m$ touches the red constraint curve are only for $m \leq m_0$. Hence, those two points of tangency are constrained local maxima of f. For similar reasons, the points of tangency on the orange and blue level curves of f are constrained local minima of f.

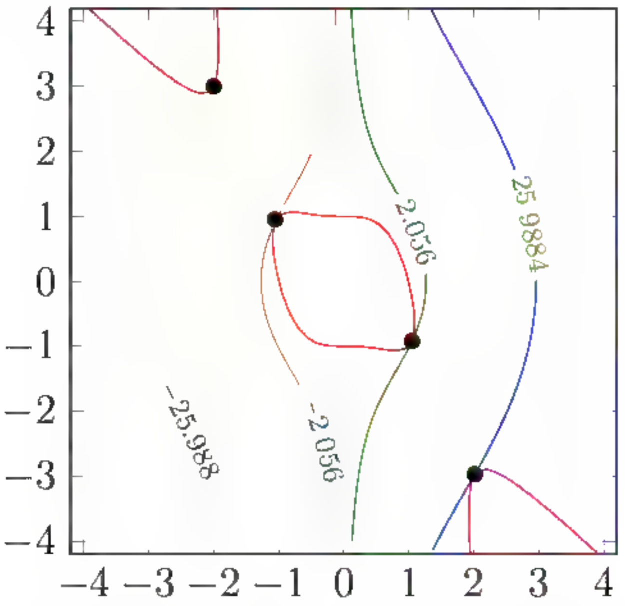


FIGURE 12.2.5. Red constraint curve $C = \{x^4 + x^3y + y^2 = 1\}$ and various colored level curves of $f(x, y) = x^3 + xy^2$ having a point (black dot) of tangential contact with C. Such points of contact with the yellow and green curves are constrained local maxima for f, and the ones on the orange and blue curves are constrained local minima for f.

■

Remark 12.2.14. The multiplier λ often has an interpretation in specific applications. Example 12.2.16 is an economics application where it corresponds to what economists call "marginal cost", and in Example 19.4.3 – a 3-constraint situation in chemistry with 3 multipliers – two of the multipliers encode temperature and pressure. The economist Paul Samuelson (1968 Nobel Prize) wrote that "market prices, regarded as parameters by perfect competitors, are nothing more nor less than Lagrange multipliers" [Sam, p. 231].

In the single-constraint setting we have been studying, there is a perspective on the meaning of λ that unifies *all* interpretations. To explain this, imagine allowing the constraint value c to vary; this is natural to do because in practice one needs to solve constrained optimization problems for *all possible* c since we may need to be ready to implement the solution for a variety of values of c (because at the outset we don't yet know what the specific value is). For instance, c may be the energy in a physical system, or the wealth in an investment portfolio.

As c varies, we expect the location $\mathbf{a}$ where f attains its maximum (or minimum) along the constraint region $g(\mathbf{x}) = c$ to move, and hence we expect the Lagrange multiplier λ to possibly change too (assuming we are not in the special case where $(\nabla g)(\mathbf{a}) = \mathbf{0}$, a case we will ignore here since it generally doesn't arise in practice). That is, we may regard $\mathbf{a}$ as a vector-valued function $\mathbf{a}(c)$ of the choice of c, and λ as a scalar-valued function $\lambda(c)$ of the choice of c. In particular, the maximal (or minimal) value $f(\mathbf{a})$ subject to the constraint region $g(\mathbf{x}) = c$ may be regarded as a function $M_f(c)$ of c.

The universal interpretation of the Lagrange multiplier in the single-constraint setting is this:

$$\lambda(c) = \frac{d}{dc}(M_f(c)). \tag{12.2.10}$$

In words, this says that the Lagrange multiplier is the rate of change with respect to c of the maximal (or minimal) value of f along the constraint region $g(\mathbf{x}) = c$ when we allow the choice of c to vary. The derivation of this formula rests on the "multivariable Chain Rule", which we will introduce in Chapter 17, so we will take up the justification of this formula there (see Example 17.1.3).

We now verify (12.2.10) directly in two situations.

Example 12.2.15. In the setting of Example 12.2.7 we were maximizing a specific function $u(x_1, x_2)$ subject to a specific constraint $g(x_1, x_2) = w$, and we found a unique point at which the maximum occurs. Plugging this point into u gives that the maximal value $M_u(w)$ for u under the constraint $g = w$ is

$$M_u(w) = u(w/3p_1, 2w/3p_2) = (w/3p_1)^{1/3}(2w/3p_2)^{2/3} = \frac{2^{2/3}}{3}\frac{w}{p_1^{1/3}p_2^{2/3}}. \tag{12.2.11}$$

The multiplier λ can also be computed, by combining our expressions for λ using the first or second equation in (12.2.2) with the determination of (x_1, x_2) via (12.2.3):

$$\lambda = \frac{1}{3p_1}\left(\frac{w}{3p_1}\right)^{2/3}\left(\frac{2w}{3p_2}\right)^{2/3} = \frac{2^{2/3}}{3}\frac{1}{p_1^{1/3}p_2^{2/3}}.$$

This expression coincides with $\dfrac{d}{dw}M_u(w)$, as we see by inspecting the explicit formula (12.2.11). ■

Example 12.2.16. To give another context in which (12.2.10) is illustrated, for given numbers $p_1, p_2 > 0$ ("prices") let's minimize a linear function $f(x_1, x_2) = p_1x_1 + p_2x_2$ subject to the constraint $g(x_1, x_2) = x_1^{1/4}x_2^{1/4} = q > 0$ (with $x_1, x_2 > 0$). In applications, x_1 and x_2 are quantities of two inputs (such as labor and capital), with respective prices of p_1 and p_2 dollars per unit of input, and g is the quantity of some product ("output") made from the inputs; here, q is a desired quantity of output for which we want to minimize the input cost $f(x_1, x_2)$. Since

$$\nabla g = \begin{bmatrix} (1/4)x_1^{-3/4}x_2^{1/4} \\ (1/4)x_1^{1/4}x_2^{-3/4} \end{bmatrix}$$

is nonzero everywhere (ruling out the first option in (12.2.1)) and $\nabla f = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$ (a constant function), the Lagrange multiplier condition $\nabla f = \lambda \nabla g$ at a minimum says that for some unknown scalar λ,

$$\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \frac{\lambda}{4}\begin{bmatrix} x_1^{-3/4}x_2^{1/4} \\ x_1^{1/4}x_2^{-3/4} \end{bmatrix}.$$

Once again "solving for λ" via the equalities among corresponding vector entries, we get

$$4p_1x_1^{3/4}x_2^{-1/4} = \lambda = 4p_2x_1^{-1/4}x_2^{3/4}.$$

Equating left and right sides, we get

$$\frac{p_1}{p_2} = \frac{x_2^{3/4}}{x_1^{1/4}}\frac{x_2^{1/4}}{x_1^{3/4}} = \frac{x_2}{x_1}.$$

Hence, $p_1x_1 = p_2x_2$, so $x_2 = (p_1/p_2)x_1$. Having solved for x_2 in terms of x_1, we need to figure out the possible values for x_1. What to do? Always remember the constraint! We have $q = x_1^{1/4}x_2^{1/4} = x_1^{1/4}((p_1/p_2)x_1)^{1/4} = (p_1/p_2)^{1/4}x_1^{1/2}$, so $x_1 = q^2\sqrt{p_2/p_1}$ and hence $x_2 = (p_1/p_2)x_1 = q^2\sqrt{p_1/p_2}$. Thus, the minimum for f under the constraint $g = q$ occurs at the point

$$\left(q^2\sqrt{p_2/p_1}, q^2\sqrt{p_1/p_2}\right).$$

At this point the minimal value for f is

$$M_f(q) = p_1x_1 + p_2x_2 = 2q^2\sqrt{p_1p_2},$$

and the multiplier λ is $4p_1(q^2\sqrt{p_2/p_1})^{3/4}(q^2\sqrt{p_1/p_2})^{-1/4} = 4p_1q^{3/2-1/2}(p_2/p_1)^{3/8+1/8} = 4p_1q\sqrt{p_2/p_1} = 4q\sqrt{p_1p_2}$. This latter expression coincides with $\frac{d}{dq}M_f(q)$ by inspection. Roughly speaking, an increase in one unit of the output (i.e., an increase in q by 1 unit) causes the minimal price $M_f(q)$ to increase approximately by $\frac{d}{dq}M_f(q) = \lambda(q)$; economists call this latter increase the "marginal cost", and we have seen that it is identified with a Lagrange multiplier. ■

Example 12.2.17. Here is an example that is more of a consistency check with things we already know than something to which we would apply the method of Lagrange multipliers in practice. For a single-variable function $F(x)$, define $f(x,y) = y$ and $g(x,y) = y - F(x)$. Then the constraint curve $g(x,y) = 0$ is the graph $y = F(x)$, so seeking local extrema for $f(x,y)$ on this curve is the problem of seeking local extrema for $F(x)$. We know that the vanishing of F' is a way to identify candidates x for such local extrema. Let's see that the method of Lagrange multipliers applied in this case yields the same criterion.

We compute $(\nabla g)(x,y) = \begin{bmatrix} -F'(x) \\ 1 \end{bmatrix}$, which is non-vanishing everywhere since its second entry is the constant function 1 (this rules out the first option in (12.2.1)), and $\nabla f = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ everywhere. Hence, a local extremum point (a,b) for f on $g = 0$ must be a point of the form $(a, F(a))$ for which $(\nabla f)(a,b) = \lambda(\nabla g)(a,b)$ for some unknown scalar λ. This says

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} -F'(a) \\ 1 \end{bmatrix} = \begin{bmatrix} -\lambda F'(a) \\ \lambda \end{bmatrix},$$

so comparing corresponding entries says $0 = -\lambda F'(a)$ and $1 = \lambda$. Plugging the second of these into the first then yields "$F'(a) = 0$" as the way to pick out candidates a for local extrema of F, exactly the condition we know from single-variable calculus. ■

Remark 12.2.18. When seeking local extrema for $f : \mathbf{R}^n \to \mathbf{R}$, we have introduced the concept of "critical point" and in Chapter 26 we will introduce a multivariable second derivative test that gives a way to check if a critical point is a local maximum or local minimum. Since the Lagrange multiplier equation is the substitute for the notion of critical point in the context of *constrained* optimization, it is natural to wonder if there is a version of the multivariable second derivative test that can be used to check if a solution to the Lagrange multiplier equation is a local maximum or local minimum on the constraint region. There is indeed such a result, but its formulation and intuitive explanation are beyond the scope of this book.

12.3. Lagrange multipliers as prices. In the economy, one way that constraints are enforced is through extra charges. Cell phone companies constrain data usage by charging extra once the user goes above their limit; airlines charge extra when a customer goes over the luggage limit.

For example, suppose you are running an airline. You would like to enforce that the average amount of check-in luggage, per passenger, is no more than 30 pounds; otherwise you run out of space. To arrange this, you charge passengers for check-in luggage, and then you *adjust the price* until the passengers average 30 pounds each.

> We can think of Lagrange multipliers as a mathematical formulation of exactly this: *enforce a constraint by charging for its failure, and adjusting the price.* (You will find this topic discussed in more detail in the economics literature under the name of *shadow prices.*)

Namely, suppose we want to find an extremum of f subject to the constraint $g = c$. We will "charge" g for going above c by maximizing not f, but instead

$$F(x,y) = f(x,y) - \lambda(g(x,y) - c)$$

where λ is the extra charge for excess g. (Warning: λ can actually be negative too, and this corresponds to penalizing g for going below c. Don't worry about it: this possibility will just come out of the final equations.) To find extremum points of $F(x,y)$, we set its gradient to be $\mathbf{0}$ to obtain

$$(\nabla F)(x,y) = \nabla f - \lambda \nabla g = \mathbf{0}, \text{ or equivalently } (\nabla f)(x,y) = \lambda(\nabla g)(x,y).$$

For any given value of λ, we can solve this equation to find the critical points (x,y) (these will generally depend on λ in some way). These critical points usually will not satisfy our original constraint $g(x,y) = c$. To force them to satisfy the constraint – just like the airline – we need to adjust λ. So in summary, we want to find a value of λ for which the pair of conditions

$$(\nabla f)(x,y) = \lambda(\nabla g)(x,y) \text{ and } g(x,y) = c$$

on (x,y,λ) can be both solved at the same time. This is exactly the same system of equations we found earlier, with Lagrange multipliers (except that the option $\nabla g = \mathbf{0}$ is avoided; in such applications ∇g is generally non-vanishing on the constraint region $g = c$)!

Example 12.3.1. Let's maximize $x^{1/4}y^{1/2}$ subject to the constraint $x + y = 3/4$ with $x, y > 0$. This is an instance of the Cobb–Douglas model from economics (discussed in Remark 12.2.8).

According to the recipe we just discussed (with $f(x,y) = x^{1/4}y^{1/2}$, $g(x,y) = x + y$), we want to find the point(s) where

$$P(x,y) = x^{1/4}y^{1/2} - \lambda(x + y - 3/4)$$

attains an extremum on the region $x, y > 0$ and then *choose* λ so that such point(s) lie on $x + y = 3/4$.

The local extrema are attained at points where $\nabla P = 0$, or upon working out the algebra:

$$\begin{bmatrix} (1/4)x^{-3/4}y^{1/2} - \lambda \\ (1/2)x^{1/4}y^{-1/2} - \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

This forces $\lambda > 0$, and by solving we get

$$(x,y) = (1/(64\lambda^4), 1/(32\lambda^4)).$$

Of course, all we really know is that this is a critical point of P (in fact, the only one), not that it is a local extremum of P. It turns out that this is a local maximum, and even an overall maximum.

There is a graphical method that can be used to affirm that this is an overall maximum, similar to the pictures illustrating Examples 12.2.4 and 12.2.12, but we won't get into that approach here since it doesn't help for problems involving more than 3 variables. In single-variable calculus, to *systematically* determine when a critical point is a local extremum and more specifically is a local maximum or local minimum, we generally use a version of the second derivative test. In Chapter 26 we will develop a version of the second derivative test that works for optimization problems in *any* number of unknowns, and in Example 26.2.5 we will revisit the present situation to affirm the maximum property via those systematic techniques.

Finally, we choose the "price" λ to enforce the constraint $x+y=3/4$. This says

$$\frac{1}{32\lambda^4}+\frac{1}{64\lambda^4}=\frac{3}{4}.$$

Solving for λ^4 yields $\lambda^4=1/16$, so $(x,y)=(1/4,1/2)$. ■

Now let's go through why Example 12.3.1 worked: why is $(1/4,1/2)$ really a maximum of f subject to the constraint $x+y=3/4$? Suppose there were some other point – (a,b) say – on the line $x+y=3/4$ (so $a+b=3/4$) but with $f(a,b)>f(1/4,1/2)$. At the points (a,b) and $(1/4,1/2)$ where the constraint $x+y=3/4$ holds, so the term $x+y-3/4$ vanishes, this says $P(a,b)>P(1/4,1/2)$. But that can't be, because $(1/4,1/2)$ is the point at which P takes its maximal value. This type of reasoning always works:

> *if $(a(\lambda),b(\lambda))$ is an unconstrained extremum of $f(x,y)-\lambda(g(x,y)-c)$ for some value of λ then it is also a constrained extremum of f on the level set of g through $(a(\lambda),b(\lambda))$.*

We then *choose* the value of λ to make sure that $g(a(\lambda),b(\lambda))=c$! To summarize:

Alternative way to think about Lagrange multipliers: to find constrained maxima (resp. minima) of f subject to $g=c$, do the following:

- Introduce an extra variable λ and maximize (resp. minimize) the function
$$F(x,y)=f(x,y)-\lambda(g(x,y)-c)$$
in x,y; note that $F=f$ on the curve $g(x,y)=c$. This maximum (resp. minimum) will generally depend on λ. Think of λ as the price of g exceeding c (resp. falling short of c).
- Finally, *choose a value of* λ for which the maximum lies on the desired level set $g=c$.

In words, to enforce $g=c$, charge a price λ per unit of g, and adjust the price suitably.

The idea behind this alternative viewpoint is really very important. It is this:

Optimization Principle. Rather than solve a constrained problem, penalize deviations from the constraint via a related function (involving a parameter – such as λ above) that you maximize or minimize without constraints.

12.4. Two more advanced contexts. We now consider two topics whose proper development entails a more advanced use of the methods we have been discussing.

Context I: multiple constraints. In realistic problems, one often encounters multiple constraints. For two constraints we use *two* Lagrange multipliers instead of one, and similarly for more constraints. We now work through a typical two-constraint example using the Optimization Principle from the end of Section 12.3. (The method with any number of constraints requires more linear algebra, so we postpone it to Section 19.4 and illustrate it in Example 19.4.3 with *three* Lagrange multipliers that arise in chemistry and Example 19.4.4 with building a linear classifier in machine learning.)

Example 12.4.1. Suppose we want to maximize $f(p,q,r) = -p\ln(p) - q\ln(q) - r\ln(r)$ subject to the constraints $p+q+r=1$ and $2p+3q+5r=4$ with $p,q,r>0$.

> This type of extremum problem arises in thermodynamics (when maximizing entropy, in accordance with the Principle of Maximum Entropy) and, done in general, it leads to the Boltzmann distribution which describes the statistical behavior of a physical system whose microscopic parts can switch between different states (such as a molecule that can switch between configurations, or a molecule that can rotate around different axes). The function f as above arises when modeling a system with three states having respective energies $2, 3, 5$ (in some units), and the values p, q, r are the probabilities of the system being in each of the three respective states. The constraint $p+q+r=1$ expresses that the system must be in one of the three states (total probability is 1), and the constraint $2p+3q+5r=4$ encodes conservation of energy.

To do this, we introduce the new function

$$F(p,q,r) = (-p\ln(p) - q\ln(q) - r\ln(r)) - \lambda(p+q+r-1) - \lambda'(2p+3q+5r-4)$$

with λ and λ' serving as the two Lagrange multipliers. Now we get

$$\frac{\partial F}{\partial p} = -\ln(p) - 1 - \lambda - 2\lambda', \quad \frac{\partial F}{\partial q} = -\ln(q) - 1 - \lambda - 3\lambda', \quad \frac{\partial F}{\partial r} = -\ln(r) - 1 - \lambda - 5\lambda'$$

We set each partial derivative equal to 0 and thereby get

$$p = e^{-(\lambda+1)}e^{-2\lambda'}, \quad q = e^{-(\lambda+1)}e^{-3\lambda'}, \quad r = e^{-(\lambda+1)}e^{-5\lambda'}. \tag{12.4.1}$$

This has solved the extremum problem *in terms of the "Lagrange multipliers"* λ and λ'.

Our first constraint "$p+q+r=1$" says $e^{-(\lambda+1)}(e^{-2\lambda'} + e^{-3\lambda'} + e^{-5\lambda'}) = 1$, or equivalently

$$e^{-(\lambda+1)} = \frac{1}{e^{-2\lambda'} + e^{-3\lambda'} + e^{-5\lambda'}}.$$

Plugging this into (12.4.1) therefore yields

$$p = \frac{e^{-2\lambda'}}{e^{-2\lambda'} + e^{-3\lambda'} + e^{-5\lambda'}}, \quad q = \frac{e^{-3\lambda'}}{e^{-2\lambda'} + e^{-3\lambda'} + e^{-5\lambda'}}, \quad r = \frac{e^{-5\lambda'}}{e^{-2\lambda'} + e^{-3\lambda'} + e^{-5\lambda'}}. \tag{12.4.2}$$

This has removed the role of the "Lagrange multiplier" λ, but λ' remains. The constraint $2p+3q+5r=4$ will allow us to find the "right" value of λ'. It tells us (after adding fractions) that

$$\frac{2e^{-2\lambda'} + 3e^{-3\lambda'} + 5e^{-5\lambda'}}{e^{-2\lambda'} + e^{-3\lambda'} + e^{-5\lambda'}} = 4.$$

This expands to

$$2e^{-2\lambda'} + 3e^{-3\lambda'} + 5e^{-5\lambda'} = 4e^{-2\lambda'} + 4e^{-3\lambda'} + 4e^{-5\lambda'},$$

or equivalently $2e^{-2\lambda'} + e^{-3\lambda'} - e^{-5\lambda'} = 0$ (note the minus sign). Entering this into WolframAlpha yields $\lambda' \approx -0.4196$, and plugging that back into (12.4.2) gives the (approximate) solution

$$p \approx 0.1655, \quad q \approx 0.2517, \quad r \approx 0.5827. \tag{12.4.3}$$

The physical context explains that this has to be a maximum, but mathematically we haven't actually justified that this really is a maximum (rather than a minimum or a saddle point, for example). To do so mathematically requires a multivariable version of the second derivative test, much as in single-variable calculus one generally has to use the second derivative test to distinguish local maxima from local minima. We will come back to this example to apply the multivariable second derivative test in Example 26.2.4. ■

Context II: linear programming, interior-point methods.

Businesses routinely face complicated optimization problems that go beyond the setting of Example 12.1.1 due to having rather more involved constraint conditions. This leads to questions addressed by a huge field of applied mathematics called *linear programming*; it arises in economics, the theory of scheduling (such as for airlines, traveling salespeople, etc.), various parts of engineering, and elsewhere. Linear programming tackles optimizing *linear* functions, such as

$$f(x_1, \ldots, x_n) = 3x_1 - 7x_2 + \cdots + 8x_n, \tag{12.4.4}$$

on special types of regions in $\mathbf{R}^n$, typically with large n. Here is an example of how a linear programming problem arises.

Example 12.4.2. Suppose your company builds a certain product at several locations (A, B, C), and needs to deliver the product to some other locations (D, E). You must determine how much of the product to ship along each of the routes AD, BD, CD, AE, BE, CE. The transportation costs along each route will be different, and the supply at each of A, B, C may also differ.

To formulate this mathematically, let us choose some numbers. Suppose the cost of shipping along routes AD, BD, CD, AE, BE, CE is (respectively) $2, 2, 3, 3, 4, 4$ dollars per unit of product. Suppose that you produce 10 units of the product at A, 20 units at B, and 40 units at C, and from those 70 units of product you need to deliver 35 units to each of D and E. Write x_{AD} for the amount of product shipped from A to D, and so on. Our constraints are then as follows:

$$x_{AD} + x_{BD} + x_{CD} = 35 \text{ (deliver 35 units to } D), \quad x_{AE} + x_{BE} + x_{CE} = 35 \text{ (deliver 35 units to } E),$$
$$x_{AD} + x_{AE} = 10, \quad x_{BD} + x_{BE} = 20, \quad x_{CD} + x_{CE} = 40 \text{ (\# units produced at } A, B, C),$$
$$x_{AD}, x_{BD}, x_{CD}, x_{AE}, x_{BE}, x_{CE} \geq 0 \text{ (e.g., can only deliver from } A \text{ to } E \text{, not vice versa)}$$

Subject to all of these constraints – which involve both linear equations and *inequalities* – we want to minimize the transportation cost, which is $2x_{AD} + 2x_{BD} + 3x_{CD} + 3x_{AE} + 4x_{BE} + 4x_{CE}$. ■

If we take the approach of solving a linear programming problem by seeking critical points then we don't get very far since any partial derivative of a linear function is constant (and usually not 0)! For example, with the function f in (12.4.4) we have $f_{x_2} = -7$. Thus, if we try to solve the simultaneous conditions

$$f_{x_1} = 0, \quad f_{x_2} = 0, \quad \ldots, \quad f_{x_n} = 0$$

we will typically find no simultaneous solutions at all. This seems to indicate that linear functions of n variables have no local maxima or local minima. Indeed, that is the case if you look at such a function on the *entirety* of $\mathbf{R}^n$ (or at interior points of any region D in $\mathbf{R}^n$). For example, with $n = 1$ consider the function $f(x) = 3x$: its graph has no peaks or valleys!

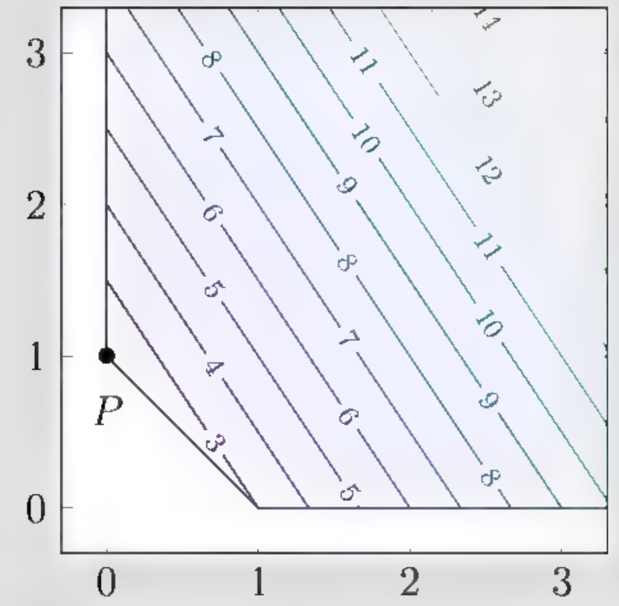
The role of linear programming, however, is to study linear functions on regions D defined by "linear inequalities" (i.e., inequalities involving linear functions), and for this restricted optimization problem the situation is more interesting.

Example 12.4.3. Consider the function $f(x, y) = 3x + 2y$ on the region

$$D = \{(x, y) \in \mathbf{R}^2 : x \geq 0,\ y \geq 0, x + y \geq 1\}$$

in the plane $\mathbf{R}^2$. This is illustrated in Figure 12.4.1.

Does f attain a global maximum or minimum on this region? It should be plausible that $f(x, y) = 3x + 2y$ gets arbitrarily large as we move out toward "infinity" in D; i.e., as we let $|x|$ or $|y|$ become arbitrarily large). On the other hand, because x and y are bounded below on D (each is ≥ 0 on D, for example) and the coefficients are ≥ 0, f is bounded below on D (in fact ≥ 0); theoretical reasons then ensure that minimum exists; how do we find it?



The function $f(x, y) = 3x + 2y$
achieves a minimum value of 2
at the point P on the boundary of D.

FIGURE 12.4.1. The region D (shown in blue) and a contour plot of $f(x, y) = 3x + 2y$.

If we follow our standard procedure, we first compute the partial derivatives. As we have observed above, these never vanish in the interior of D. So there are no interior critical points. This implies that a global minimum *must* be on the boundary. Now restrict f to any one of the three boundary curves: the two half-lines

$$\{(0, y) : y \geq 1\}, \ \ \{(x, 0) : x \geq 1\}$$

and the line segment

$$\{(x, y) : x + y = 1, \ x, y \geq 0\} = \{(x, 1 - x) : 0 \leq x \leq 1\}.$$

We claim that the *linear* $f(x, y) = 3x + 2y$ cannot attain an extremum in the interior of any of these boundary curves (i.e., away from their endpoints). The reason is that the restriction of f to each of these boundary curves is a *linear* function (of the type $at + b$ for some $a, b \in \mathbf{R}$, with t a parameter along the boundary curve) of 1 variable: on the two half-lines we get $f(0, y) = 2y$ and $f(x, 0) = 3x$, and on $x + y = 1$ we have $f(x, 1 - x) = 3x + 2(1 - x) = 2 + x$.

This leaves as the only *possible* place where f can attain a global minimum the two corner points $(1, 0)$ and $(0, 1)$ where boundary curves meet. We compute $f(1, 0) = 3$ and $f(0, 1) = 2$. Based on all that we have done, we conclude that the global minimum of f on D is the value 2, attained at the corner point $P = (0, 1)$ on the boundary of D, as shown in Figure 12.4.1. ■

Remarkably, *many algorithms to solve linear programming problems use the linear algebra and optimization ideas and concepts from this course* (such as gradient descent), as we now illustrate.

Example 12.4.4. Suppose that we wish to minimize $x + y$ subject to the simultaneous constraints

$$2x + 3y \geq 1, \ \ 3x + y \geq 1, \ \ x \geq 0, \ \ \ y \geq 0. \tag{12.4.5}$$

Try to solve this by hand first by drawing a picture of the region defined by (12.4.5) and seeing which among the level curves $x + y = a$ (parallel lines!) touch the region (12.4.5); the smallest such a is the desired minimal value. You should be able to see by such drawing that the minimum value is attained at the corner $(x, y) = (2/7, 1/7)$, where the minimal value is $3/7$. But we shall now discuss another method that is far more robust.

Drawing pictures does not work for the far more complicated linear programming problems that are often encountered in practice, with enormous numbers of variables and constraints. So here is another technique, using gradient descent and the Optimization Principle at the end of Section 12.3: rather than try to grapple with $x + y$ directly subject to the constraints (12.4.5), we will try to "penalize" any (x, y) not satisfying those constraints. More precisely, rather than working with the function $x + y$, we will work with

$$f(x, y) = x + y + \varepsilon G(x, y)$$

where G is a "cost function" to be defined and ε is a (small positive) parameter to be chosen.

The great idea is to choose $G(x, y)$ that is "small" on most of the region (12.4.5), so on most of that region we have $f(x, y) \approx x + y$, but to ensure $G(x, y)$ "gets large" when we approach the boundary of the region (so if we try to run gradient descent beginning at a point deep within the constraint region, we won't leave the constraint region). This will be made quantitatively precise via ε. A convenient choice of $G(x, y)$ is

$$G(x, y) = -\ln(x) - \ln(y) - \ln(2x + 3y - 1) - \ln(3x + y - 1),$$

using negatives of logarithms of the expressions whose non-negativity defines the constraint region since $-\ln(t)$ is very large for positive t close to 0. For example, when x approaches 0 from above, the term $-\ln(x)$ becomes large, and so $G(x, y)$ becomes large. (We do not go into the general theory of how to select such a function G which will behave well for computational purposes; for our current problem this particular G turns out to be a good choice from the perspective of the general theory.)

Of course, we are now minimizing a different function $f(x, y)$ rather than the original $x + y$, but if we choose ε very small then we should expect that the minimum of $f(x, y)$ will approximate the true solution to our problem. (The general theory that we are not getting into addresses this issue in a systematic way.)

For any fixed value of ε we can minimize the resulting function $f(x, y)$ (that depends on ε) using gradient descent. We have done this for $\varepsilon = 0.1$ in Example 11.3.6! Starting with $\varepsilon = 0.1$ and $(x, y) = (1, 1)$, we ran the gradient descent in Example 11.3.6, arriving at the minimizer $(0.455, 0.254)$.

We now want to do this again with ε smaller. Let's try $\varepsilon = 0.05$. One of the problems with gradient descent is that we don't always have a good guess for a starting point. But in this case we take the solution we just found $(0.455, 0.254)$ in the previous step as our starting point! Repeat gradient descent, but now for the function

$$x + y + (0.05)G(x, y)$$

We also need to choose a "step size" for the gradient descent (i.e., a value for $t < 0$ as in Example 11.3.5); for this computation and the ones below, we use $t = -(0.1)(\varepsilon)$. After 1000 iterations we get to the point $(0.374, 0.175)$. This minimizes the function $x + y + (0.05)G(x, y)$, but that's still not what we want.

Now reduce ε *again* and repeat (always using the minimizer of the previous step as the starting point for the next step). The results are as follows:

- $\varepsilon = 0.1$: $(0.455, 0.254)$ – minimizes $x + y + (0.1)G(x, y)$.
- $\varepsilon = 0.05$: $(0.374, 0.175)$ – minimizes $x + y + (0.05)G(x, y)$.
- $\varepsilon = 0.02$: $(0.328, 0.143)$ – minimizes $x + y + (0.02)G(x, y)$.
- $\varepsilon = 0.01$: $(0.309, 0.140)$ – similar.

- $\varepsilon = 0.005$: $(0.298, 0.141)$.
- $\varepsilon = 0.002$: $(0.291, 0.142)$.
- $\varepsilon = 0.001$: $(0.288, 0.142)$.
- $\varepsilon = 0.0005$: $(0.287, 0.143)$.
- $\varepsilon = 0.0002$: $(0.286, 0.143)$ – minimizes $x + y + (0.0002)G(x, y)$.

It is clear that the answer is getting closer and closer to a limiting value, and the function being minimized is becoming closer and closer to $x + y$. In fact, the answer at the final step above agrees, to three decimal places, with the true solution $(2/7, 1/7)$!

This kind of technique is very powerful in practice. But we emphasize that our example above (to minimize $x + y$ on a region) involved a very simple initial function (before we brought out the negative logarithms), and we have not grappled with the details needed to make it work in more realistic settings. In practice, a lot of attention needs to be paid to these details (where to start, how to choose G, how to reduce ε, to use gradient descent or a more sophisticated technique, etc.). ■

(The simplex algorithm). Example 12.4.3 illustrates a beautiful and remarkable principle: when we seek extrema of a *linear* function f on a region D that is bounded by pieces of lines, planes, etc., then away from exceptional circumstances (such as when an extremum is attained along an entire boundary line segment, for example), the extrema are either not attained (e.g., the function may go to $\pm\infty$ in various directions) or are attained at the *corners* of the boundary of D.

Thus, in such cases we do not need to go through any work computing partial derivatives of f and then looking at the restrictions of f to the various boundary lines, planes, etc. Instead, we can immediately jump to computing the values of f at the corners of the boundary of D and deciding which of these is the largest or smallest (and confirming f doesn't tend to $\pm\infty$ along some direction in D). This is typically a comparison of values of the linear f at *finitely many points*.

This sounds so simple that it might seem bizarre that linear programming is a serious area of study. However, the linear programming extrema problems encountered in real-world applications such as in optimal transportation scheduling ("traveling salesman problem" or airline scheduling), energy distribution across a power grid, or the routing of telecommunications often involve tens or hundreds of thousands or even millions of independent variables (so working in $\mathbf{R}^n$ with rather gigantic n), and the region D defined by linear inequalities tends to have a truly huge number of corners on the boundary (for a concept of "corner on the boundary" in $\mathbf{R}^n$ that one has to *define* for general n!).

Although we could (in principle) evaluate the linear f at all of the boundary corners and compare those finitely many values, in practice this is prohibitive and usually impossible in any reasonable amount of time. What can we do instead? There are multiple approaches. One algorithm is called the *simplex method* and works very effectively without using any calculus. Though it is very successful, it doesn't have much in common with the mathematics of this course. Another useful class of methods, called *interior point* methods, uses ideas from multivariable calculus. Although the details of such methods are beyond the scope of this course, an instance of interior point methods was given in Example 12.4.4 to illustrate how it relates to the optimization methods we are discussing.

12.5. Why does the method of Lagrange multipliers work? To explain the key idea behind Theorem 12.2.1, we first note that if $(\nabla g)(\mathbf{a}) = \mathbf{0}$ then there is nothing to do. Hence, we may and do now focus on the case $(\nabla g)(\mathbf{a}) \neq \mathbf{0}$. We also focus on the case of local maxima; the case of local minima goes the same way (or apply the case of local maxima to $-f$ in place of f).

To convey the main idea with a minimum of fuss, we shall consider the special case $n = 2$: functions f and g on $\mathbf{R}^2$, so we may visualize the level set $g(\mathbf{x}) = c$ near $\mathbf{a}$ as a curve in $\mathbf{R}^2$. (The reasoning for functions on $\mathbf{R}^n$ goes similarly when $n > 2$, but the geometry then becomes a bit more involved since the level set $g(\mathbf{x}) = c$ in $\mathbf{R}^n$ near $\mathbf{a}$ is not a curve but rather is an $(n-1)$-dimensional "hypersurface"; e.g., for $n = 3$ it is a surface in $\mathbf{R}^3$.)

By assumption, the point $\mathbf{a}$ on the level set $g(\mathbf{x}) = c$ is one at which f attains a local maximum. For instance, if we think of $f(\mathbf{x})$ as the temperature at the point $\mathbf{x}$ then $\mathbf{a}$ is a point on the curve $g(\mathbf{x}) = c$ where the temperature is at least as hot as at all nearby points on the curve.

Imagine an insect crawling along on the curve $g(\mathbf{x}) = c$ with nonzero velocity, and suppose that at time $t = 0$ it is at the point $\mathbf{a}$. If we let $p(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \in \mathbf{R}^2$ be the position of the insect at time t then $p(0) = \mathbf{a}$. Now $f(p(t))$ is the temperature of the insect's location at time t, so at time 0 the insect is at a point at least as hot as all nearby points (because that is when it is at $\mathbf{a}$.) Since the function $f(p(t))$ therefore has a local maximum at $t = 0$, by single-variable calculus we know that

$$\frac{d}{dt} f(p(t)) = 0 \qquad \text{when } t = 0. \tag{12.5.1}$$

In single-variable calculus you learned how to compute the derivative of "composite functions" of the form $h(u(t))$ for functions $u : \mathbf{R} \to \mathbf{R}$ and $h : \mathbf{R} \to \mathbf{R}$ via the "Chain Rule" formula $dh/dt = (dh/du)(du/dt)$. But $f(p(t))$ is a composition of functions $p : \mathbf{R} \to \mathbf{R}^2$ and $f : \mathbf{R}^2 \to \mathbf{R}$ where the intermediate step involves $\mathbf{R}^2$ rather than $\mathbf{R}$, so it doesn't fit that paradigm: the inner function p is vector-valued rather than scalar-valued. This is a new situation that we have never encountered before!

In Chapter 17 we will discuss a version of the Chain Rule adapted to the vector-valued setting. In the special case of composing functions $p : \mathbf{R} \to \mathbf{R}^n$ and $f : \mathbf{R}^n \to \mathbf{R}$, this yields a formula for the calculus derivative $(f \circ p)'(t)$ in terms of a dot product of a gradient vector for f against a velocity vector for p (see (17.1.2)):

$$\frac{d}{dt} f(p(t)) = ((\nabla f)(p(t))) \cdot p'(t),$$

where $p'(t)$ denotes the velocity $\begin{bmatrix} x'(t) \\ y'(t) \end{bmatrix}$ of the insect at time t. Setting t to be 0 in this gives a reformulation of (12.5.1) as:

$$(\nabla f)(\mathbf{a}) \cdot p'(0) = 0. \tag{12.5.2}$$

The nonzero velocity $p'(0)$ at time $t = 0$ is tangent to the level curve $g(\mathbf{x}) = c$ at $p(0) = \mathbf{a}$ since the insect is crawling along this level curve (and the velocity of a moving particle or insect is always tangent to the path of motion). Consequently, (12.5.2) means that $(\nabla f)(\mathbf{a})$ is *perpendicular* to the tangent line to the curve $g(\mathbf{x}) = c$ at the point $\mathbf{a}$.

The gradient vector $(\nabla g)(\mathbf{a})$ is *also* perpendicular to the same curve $g(\mathbf{x}) = c$ at the same point $\mathbf{a}$ (by Theorem 11.2.1, applied to g). Since $(\nabla g)(\mathbf{a}) \neq \mathbf{0}$, it follows that the line L through the origin that it spans is *parallel* to the direction perpendicular to the curve $g(\mathbf{x}) = c$ at $\mathbf{a}$, or in other words that it is perpendicular to the tangent line of the curve $g(\mathbf{x}) = c$ at $\mathbf{a}$. Since $(\nabla f)(\mathbf{a})$ is also perpendicular to the *same* tangent direction, it lies in the line L through the origin that is perpendicular to that tangent direction, and we have already argued that L is spanned by $(\nabla g)(\mathbf{a})$. Being in that span says $(\nabla f)(\mathbf{a}) = \lambda(\nabla g)(\mathbf{a})$ for some scalar λ.

Chapter 12 highlights (links to highlights in [illegible] and next chapters)

Notation	**Meaning**	**Location in text**
λ	Greek analogue of lower-case "el"; read as "lambda"	Theorem 12.2.1

Concept	**Meaning**	**Location in text**
constrained optimization	for $f : \mathbf{R}^n \to \mathbf{R}$ and $g : \mathbf{R}^n \to \mathbf{R}$, this is the problem of seeking (local) extrema for $f(\mathbf{x})$ subject to the constraint $g(\mathbf{x}) = c$	Example 12.1.3, Example 12.1.4
Lagrange multiplier	when optimizing $f : \mathbf{R}^n \to \mathbf{R}$ subject to $g = c$, it is the scalar λ for which $(\nabla f)(\mathbf{a}) = \lambda(\nabla g)(\mathbf{a})$ (for constrained local extremum $\mathbf{a} \in \mathbf{R}^n$ of f with $(\nabla g)(\mathbf{a}) \neq \mathbf{0}$)	Theorem 12.2.1, Figure 12.1.3

Result	**Meaning**	**Location in text**
method of Lagrange multipliers	for $f : \mathbf{R}^n \to \mathbf{R}$ and $g : \mathbf{R}^n \to \mathbf{R}$, when studying $f(\mathbf{x})$ subject to the constraint $g(\mathbf{x}) = c$ any *constrained* local extremum $\mathbf{a}$ satisfies either $(\nabla g)(\mathbf{a}) = \mathbf{0}$ or $(\nabla f)(\mathbf{a}) = \lambda(\nabla g)(\mathbf{a})$ for some (unknown!) scalar λ	Theorem 12.2.1
tangential interpretation of Lagrange multiplier condition when $n = 2$	for constrained local extrema $\mathbf{a} \in \mathbf{R}^2$ of $f : \mathbf{R}^2 \to \mathbf{R}$ at which ∇f and ∇g are nonzero, the level curves of f and g in $\mathbf{R}^2$ through $\mathbf{a}$ touch tangentially at $\mathbf{a}$	Remark 12.2.5, Figure 12.2.5

Skill	**Location in text**
visualize $(\nabla f)(\mathbf{a})$ pointing along same line as $(\nabla g)(\mathbf{a})$ for constrained local extremum $\mathbf{a}$ of f subject to $g = c$ (with $(\nabla g)(\mathbf{a}) \neq \mathbf{0}$)	Figures 12.1.3, 12.2.1
use Lagrange multiplier method to solve constrained optimization in 2 or 3 variables	Examples 12.2.9, 12.2.11, 12.2.12
set up Lagrange multiplier equations for constrained extrema (maybe too complicated to solve by hand)	Example 12.2.13 (especially (12.2.8))

Part III

Geometry and algebra of matrices

"As long as algebra and geometry have been separated, their progress has been slow and their uses limited; but when these two sciences have been united, they lent each other mutual forces, and have marched together towards perfection."

J-L. Lagrange

"The Matrix is everywhere. It is all around us."

Morpheus

Overview of Part III

This Part of the book, comprising Chapters 13-18, introduces the notion of *matrix* in algebraic and geometric ways and discusses initial applications (including to multivariable calculus). We introduce the notions of *linear transformation* and *matrix multiplication*; these provide sophisticated ways to extract information from matrices and are a ubiquitous tool in the modeling of many real-world situations (across economics, biology, computer science, physics, etc.). Much of data science requires the systematic use of the properties of matrix multiplication (collectively known as "matrix algebra"), as do many areas of probability and statistics as well as physics.

Chapter 13 introduces the fundamental concept of *linear function* $\mathbf{R}^n \to \mathbf{R}^m$ and related notion of *matrix*, and describes a general object that deserves to be called the (total) derivative of a (typically nonlinear) function $f : \mathbf{R}^n \to \mathbf{R}^m$. This (total) derivative is called the *derivative matrix*. The reasons why it is the "right" general concept and what it is good for will be the focus of the rest of Part III.

In Chapter 14 we introduce *linear transformations* as a geometric way to think about matrices. The visual idea of *composing* two linear transformations (i.e., applying one and then another, such as rotating $\mathbf{R}^2$ around the origin by an angle and then shearing it in some direction) is calculated using a new algebraic idea: "multiplying" matrices. In the context of rigid motions in 3-dimensional space (relevant in robotics, video games, and computer vision) this yields a rich array of examples, applications, and insights.

An important feature of matrices is that the product AB of matrices A and B *depends on the order of multiplication*; i.e., typically $AB \neq BA$. This phenomenon is studied further in Chapter 15, where we also introduce matrix addition and provide a list of familiar-looking properties of addition and multiplication for matrices (explaining why matrix "multiplication" really deserves to be called "multiplication").

A striking application of matrix algebra arises via powers $M, M^2, M^3, M^4, \dots$ of a single $m \times m$ matrix M (with possibly huge m), discussed in Chapter 16. In contexts as diverse as Google's PageRank algorithm (where m is more than 1 billion), the Wright-Fisher model for genetic drift, and long-term odds of winning for a casino game, the computation and/or properties of powers M^n for large n underlies the mathematical analysis. A unifying theme here is the concept of *Markov chain*, wherein there is a feedback loop sending each stage of a process into the next subject to a probabilistic rule. It is a surprising (yet useful) fact that M^n for large n behaves in accordance with predictable patterns.

In Chapter 17 we introduce a multivariable analogue of the Chain Rule from single-variable calculus. This is a formula *in terms of matrix multiplication* that reduces computing the derivative of a composition of vector-valued functions to (i) computing many single-variable derivatives and (ii) *combining those* in an appropriate manner. This "combining" step is clarified a lot by matrix algebra: it makes the multivariable Chain Rule look intuitively reasonable (without matrices it looks strange). This Chain Rule pervades rates of change in natural sciences, economics, machine learning algorithms, and artificial neural networks.

In Chapter 18 we introduce the idea of *matrix inverse*. There is no general "cancellation law" for matrix multiplication (i.e., we cannot generally "divide" by a nonzero matrix), but there is a large class of $n \times n$ matrices for which there is an analogue of division: this involves the concept of matrix inverse (recovering "$1/x$" when $n = 1$). An important calculus application of inverses of derivative matrices is the multivariable Newton's method (used in GPS, robotics, etc.), discussed at the end of Chapter 18.

13. Linear functions, matrices, and the derivative matrix

A useful notion of "derivative" at $\mathbf{a} \in \mathbf{R}^n$ for a scalar-valued function $f : \mathbf{R}^n \to \mathbf{R}$ is the gradient $(\nabla f)(\mathbf{a}) \in \mathbf{R}^n$ (e.g., the linear approximation (11.1.2) "justifies" this way of thinking about the gradient).

Seeking a suitable notion of "derivative" at $\mathbf{a} \in \mathbf{R}^n$ for a *vector-valued* (rather than scalar-valued) function $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ will lead us to an even more powerful concept: a derivative matrix.

By the end of this chapter, you should be able to:

- distinguish linear functions from more general functions;
- multiply matrices by vectors, and abbreviate a linear function using matrix-vector notation;
- compute the derivative matrix, and compute a local approximation from a derivative matrix.

13.1. The linear approximation to a vector-valued function. Suppose that $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ is a vector-valued function, with component functions $f_1, f_2, \dots, f_m$:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix}$$

for $\mathbf{x} \in \mathbf{R}^n$. For an n-vector $\mathbf{a}$, how can we approximate the m-vector $\mathbf{f}(\mathbf{x})$ when $\mathbf{x}$ is near $\mathbf{a}$? To make things easier to imagine, suppose $m = n = 3$. Let's consider a specific example.

Example 13.1.1. Define $\mathbf{g} : \mathbf{R}^3 \to \mathbf{R}^3$ by $\mathbf{g}(x, y, z) = (e^z(x-y)^2, 2yz + x^3, x^2 - y^3 + z)$, and let $\mathbf{a} = (2, 1, 0)$. For the respective component functions $g_1(x, y, z) = e^z(x-y)^2$, $g_2(x, y, z) = 2yz + x^3$, and $g_3(x, y, z) = x^2 - y^3 + z$, if (x, y, z) is near $\mathbf{a}$ then we have

$$\begin{aligned} g_1(x,y,z) &\approx g_1(\mathbf{a}) + ((\nabla g_1)(\mathbf{a})) \cdot (x-2, y-1, z) = 1 + (2, -2, 1) \cdot (x-2, y-1, z), \\ g_2(x,y,z) &\approx g_2(\mathbf{a}) + ((\nabla g_2)(\mathbf{a})) \cdot (x-2, y-1, z) = 8 + (12, 0, 2) \cdot (x-2, y-1, z), \\ g_3(x,y,z) &\approx g_3(\mathbf{a}) + ((\nabla g_3)(\mathbf{a})) \cdot (x-2, y-1, z) = 3 + (4, -3, 1) \cdot (x-2, y-1, z). \end{aligned}$$

Putting these scalar approximations together yields a vector approximation

$$\mathbf{g}(x,y,z) \approx \begin{bmatrix} 1 + (2,-2,1) \cdot (x-2, y-1, z) \\ 8 + (12,0,2) \cdot (x-2, y-1, z) \\ 3 + (4,-3,1) \cdot (x-2, y-1, z) \end{bmatrix} = \begin{bmatrix} -1 + 2x - 2y + z \\ -16 + 12x + 2z \\ -2 + 4x - 3y + z \end{bmatrix} \tag{13.1.1}$$

(the equality entailing working out the dot products and combining constant terms). The right side in (13.1.1) involves a lot of numbers! ■

More generally, still with $m = n = 3$, we want to approximate the vector $\mathbf{f}(x,y,z) = \begin{bmatrix} f_1(x,y,z) \\ f_2(x,y,z) \\ f_3(x,y,z) \end{bmatrix}$ for (x, y, z) near a point (a, b, c) for *any* scalar-valued functions f_1, f_2, f_3. We have

$$\begin{aligned} f_i(x,y,z) &\approx f_i(a,b,c) + ((\nabla f_i)(a,b,c)) \cdot (x-a, y-b, z-c) \\ &= f_i(a,b,c) + \left(\frac{\partial f_i}{\partial x}(a,b,c)\right)(x-a) + \left(\frac{\partial f_i}{\partial y}(a,b,c)\right)(y-b) + \left(\frac{\partial f_i}{\partial z}(a,b,c)\right)(z-c), \end{aligned}$$

so combining these for f_1, f_2, and f_3 yields

$$\begin{bmatrix} f_1(x,y,z) \\ f_2(x,y,z) \\ f_3(x,y,z) \end{bmatrix} \approx \mathbf{f}(a,b,c) + \begin{bmatrix} \frac{\partial f_1}{\partial x}(a,b,c)(x-a) + \frac{\partial f_1}{\partial y}(a,b,c)(y-b) + \frac{\partial f_1}{\partial z}(a,b,c)(z-c) \\ \frac{\partial f_2}{\partial x}(a,b,c)(x-a) + \frac{\partial f_2}{\partial y}(a,b,c)(y-b) + \frac{\partial f_2}{\partial z}(a,b,c)(z-c) \\ \frac{\partial f_3}{\partial x}(a,b,c)(x-a) + \frac{\partial f_3}{\partial y}(a,b,c)(y-b) + \frac{\partial f_3}{\partial z}(a,b,c)(z-c) \end{bmatrix}. \tag{13.1.2}$$

A special case of this is given in the middle of (13.1.1).

Alas, the right side of (13.1.2) is horrifying. We need a more compact and efficient way to work with it. The way to handle it is provided by the language of matrices, to which we now turn.

13.2. Linear and affine functions. Functions on the right side of (13.1.1) and (13.1.2) have a name:

Definition 13.2.1. A scalar-valued function $f : \mathbf{R}^n \to \mathbf{R}$ is called

- *affine* if it has the form $f(x_1, \dots, x_n) = a_1x_1 + a_2x_2 + \cdots + a_nx_n + b$ for some numbers $a_1, \dots, a_n, b$ (so $b = f(\mathbf{0})$).
- *linear* if it has the form $f(x_1, \dots, x_n) = a_1x_1 + a_2x_2 + \cdots + a_nx_n$ for some numbers $a_1, \dots, a_n$; i.e., it is affine with $b = 0$, or equivalently with $f(\mathbf{0}) = 0$.

A vector-valued function $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ (so $f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$) is called

- *affine* if each of its component functions $f_i : \mathbf{R}^n \to \mathbf{R}$ is affine.
- *linear* if each of its component functions $f_i : \mathbf{R}^n \to \mathbf{R}$ is linear.

An example of an affine vector-valued function is the total right side of (13.1.1). Here are some more:

Example 13.2.2. The function $\mathbf{f}(x,y,z) = \begin{bmatrix} x-y+z+3 \\ z-x \\ y+x+1 \end{bmatrix} = \begin{bmatrix} x-y+z+3 \\ -x+0y+z \\ x+y+0z+1 \end{bmatrix}$ from $\mathbf{R}^3$ to $\mathbf{R}^3$ is affine. The function $\mathbf{g}(x,y,z) = \begin{bmatrix} x-y+z \\ z-x \\ y+x \end{bmatrix}$ from $\mathbf{R}^3$ to $\mathbf{R}^3$ is linear, and $\mathbf{h}(x,y,z) = \begin{bmatrix} x-y+z \\ z-x \\ y+x+z^2 \end{bmatrix}$ is neither affine nor linear (due to z^2 in the third component function $h_3 = y + x + z^2$). ■

Warning. In the case of a single variable ($n = 1$), the affine scalar-valued functions are those of the form $f(x) = ax + b$ and the linear ones are the affine ones with $b = 0$. This differs from what is done in high school, where "linear" refers to $ax + b$ (allowing $b \neq 0$), but in all uses of linear algebra the meaning of "linear" for a function $f : \mathbf{R}^n \to \mathbf{R}$ is universally understood to be what we define above (with no constant term). Although affine functions arise in some situations, it is the linear functions that play a particularly big role in linear algebra, and so we shall focus more on them. For instance, in Example 8.1.4, addition is linear but multiplication is not. The function in Example 8.1.5 is not linear, but both functions in Example 8.1.6 are linear. Finally, the functions in Example 8.1.12 and Example 8.1.13 are both linear.

13.3. Matrices: a shorthand for linear functions. From now on, we denote vector-valued functions with either of the fonts f or $\mathbf{f}$; *context should make clear when "f" is vector-valued or scalar-valued.* Let us write out, in full glory, the definition of a linear function from $\mathbf{R}^3$ to $\mathbf{R}^3$. A function $f : \mathbf{R}^3 \to \mathbf{R}^3$ is linear "if each of its component functions is linear." In other words, this means that

$$f\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} ax+by+cz \\ dx+ey+fz \\ gx+hy+iz \end{bmatrix}, \tag{13.3.1}$$

where $a, b, c, d, e, f, g, h, i$ are real numbers. For instance, with $\mathbf{g}$ as in Example 13.1.1 and (x, y, z) near $(2, 1, 0)$ we saw in (13.1.1) that

$$\mathbf{g}\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) \approx \begin{bmatrix} -1+2x-2y+z \\ -16+12x+2z \\ -2+4x-3y+z \end{bmatrix} = \begin{bmatrix} -1 \\ -16 \\ -2 \end{bmatrix} + \begin{bmatrix} 2x-2y+z \\ 12x+0y+2z \\ 4x-3y+z \end{bmatrix}.$$

This is an approximation with an affine function, the bulk of which is a linear function (disregarding the addition against $(-1, -16, -2)$).

As you can see, there are a lot of numbers to keep track of! To organize such information, the following notion will be very convenient.

Definition 13.3.1. An *$m \times n$ matrix* is a rectangular array A of numbers presented like this:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{bmatrix}.$$

The collection of entries $\begin{bmatrix} a_{i,1} & a_{i,2} & \dots & a_{i,n} \end{bmatrix}$ along the ith horizontal layer (with $i = 1$ along the top side) is called the ith *row*, and the collection of entries $\begin{bmatrix} a_{1,j} \\ a_{2,j} \\ \vdots \\ a_{m,j} \end{bmatrix}$ along the jth vertical layer (with $j = 1$ along the left side) is called the jth *column*.

One way to remember which is a "row" and which is a "column" is that theaters and airplanes have "rows" of seats arranged horizontally (left/right) whereas buildings in ancient Rome had "columns" that are vertical (up/down). By visualizing the array of numbers, in an $m \times n$ matrix we see columns have m entries and rows have n entries. Don't memorize things like this; you'll learn it via examples.

The entry at the crossing of the ith row from the top and jth column from the left is denoted a_{ij} (or sometimes $a_{i,j}$); it is called the *ij-entry* or *(i, j)-entry*. (The convention that the first index i labels the row position starting from the top and the second index j labels the column position starting from the left is due to the historical origin of matrices in work with systems of linear equations and pervades formulas for applications linear algebra in *all* fields.[14] You will get accustomed to it with experience.)

Example 13.3.2. Some examples of matrices are

$$A = \begin{bmatrix} 6 & 11 & -5 \\ 1 & 1 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} -2 & 0 \\ 3 & 1 \\ 7/2 & -1 \end{bmatrix}, \quad C = \begin{bmatrix} \sqrt{3}/2 & 1/2 \\ -1/2 & \sqrt{3}/2 \end{bmatrix}.$$

The matrix A is a 2×3 matrix, whereas B is a 3×2 matrix, and C is a 2×2 matrix. We have $a_{13} = -5$ and $b_{31} = 7/2$. (In Example 14.4.2 we will see that C naturally arises when describing rotation around the origin by 30 degrees clockwise.)

Since we'll use matrices throughout the rest of the book, look again at A and B to make sure you can recognize what is meant by "2×3 matrix" (such as A) and what is meant by "3×2 matrix" (such as B), so that when someone says "$m \times n$ matrix" you will remember it means m layers horizontally and n layers vertically. Make sure you recognize which is b_{12} and which is b_{21}, and that for the specific B as above the expression b_{32} makes sense but b_{23} does not (there is no 3rd entry in the 2nd row)! ■

---

[14] In computer graphics, `Pixel(i,j)` has the "flipped" meaning of referring to color intensity of the pixel in the ith column from the left and jth row from the top in a rectangular array of pixels, where moreover i and j each begin at 0 rather than at 1. But this has nothing to do with linear algebra; for *all* uses of linear algebra in computer science, economics, physics, engineering, statistics, etc., our definition of $a_{i,j}$ is what is universally used. For an analogous oddity in economics, look here

Remark 13.3.3. In practice, one usually writes "ij-entry" rather than "(i, j)-entry". The former may cause confusion when specific numbers beyond single digits are used for i or j (as happens if m or n is at least 10), such as $i = 37$ and $j = 4$: what does "374-entry" mean? Row 37 and column 4, or row 3 and column 74? In situations where such confusion may arise, the notation "(i, j)-entry" is used instead (and we may write $a_{37,4}$ to distinguish it from $a_{3,74}$, and speak of the $(37, 4)$-entry and the $(3, 74)$-entry).

Definition 13.3.4. If A is an $m \times n$ matrix, and $\mathbf{x} \in \mathbf{R}^n$, the *matrix-vector product* $A\mathbf{x} \in \mathbf{R}^m$ is defined as

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{bmatrix}.$$

In other words, if we write $\mathbf{r}_1, \dots, \mathbf{r}_m$ for the *rows* of A (so these are n-vectors), then

$$A\mathbf{x} = \begin{bmatrix} \mathbf{r}_1 \cdot \mathbf{x} \\ \mathbf{r}_2 \cdot \mathbf{x} \\ \vdots \\ \mathbf{r}_m \cdot \mathbf{x} \end{bmatrix}.$$

WARNING: if A is an $m \times n$ matrix and $\mathbf{x}$ is a d-vector with $d \neq n$ then the matrix-vector product $A\mathbf{x}$ is *not defined*. If you ever find yourself writing such a product with $d \neq n$ then you have made a mistake; always check that your matrix-vector products *make sense* before computing anything.

Let's use this notation in examples, to see that matrix-vector products encode linear functions.

Example 13.3.5. The linear function $g : \mathbf{R}^3 \to \mathbf{R}^3$ from Example 13.2.2 is given by

$$g\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

Work out the matrix-vector product on the right side to check for yourself that this really holds. ■

Example 13.3.6. For f as in (13.3.1), we have

$$f\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

■

Example 13.3.7. For $\mathbf{g}$ as in Example 13.1.1 and (x, y, z) near $(2, 1, 0)$ we have

$$\mathbf{g}\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) \approx \begin{bmatrix} 1 \\ 8 \\ 3 \end{bmatrix} + \begin{bmatrix} 2 & -2 & 1 \\ 12 & 0 & 2 \\ 4 & -3 & 1 \end{bmatrix} \begin{bmatrix} x-2 \\ y-1 \\ z \end{bmatrix} = \begin{bmatrix} -1 \\ -16 \\ -2 \end{bmatrix} + \begin{bmatrix} 2 & -2 & 1 \\ 12 & 0 & 2 \\ 4 & -3 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

(the equality entails expanding out each entry of the first matrix-vector product to extract a total constant term for each entry, as we did in (13.1.1)). ■

More generally, if you go back to the definitions, you'll see:

Proposition 13.3.8. A function $f : \mathbf{R}^n \to \mathbf{R}^m$ is linear precisely when $f(\mathbf{x}) = A\mathbf{x}$ for an $m \times n$ matrix A.

This just rephrases the *definition* of "linear function". In this way, an $m \times n$ matrix A is a shorthand way of encoding a linear function from $\mathbf{R}^n$ to $\mathbf{R}^m$ (*not* the other way around from $\mathbf{R}^m$ to $\mathbf{R}^n$ – when $m \neq n$ – since $A\mathbf{x}$ only makes sense for $\mathbf{x} \in \mathbf{R}^n$).

Example 13.3.9. In Example 8.1.12 we described a function that rotates vectors by 90 degrees counterclockwise, namely $T : \mathbf{R}^2 \to \mathbf{R}^2$ given by

$$T\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} -y \\ x \end{bmatrix}.$$

This geometrically-defined function is given by a matrix-vector product too (so it is linear):

$$T\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 0 \cdot x + (-1) \cdot y \\ 1 \cdot x + 0 \cdot y \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

Thus, T arises from the 2×2 matrix $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. ■

Example 13.3.10. We can use matrix-vector products to encode *systems* of linear equations. In Example 10.3.1, we encountered the system of equations

$$-15m + 5b = 0, \quad 55m - 15b = 12.$$

In matrix notation, this becomes: $\begin{bmatrix} -15 & 5 \\ 55 & -15 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ 12 \end{bmatrix}.$ ■

Example 13.3.11. A graduate student realizes after several months of living on a diet of Ramen noodles that this is not a way to get the recommended daily intake of vitamins. So the student decides to get daily nutrition from two food sources rather than one! Say the two food sources are called F_1 and F_2, with F_1 costing 70 cents per ounce and F_2 costing 30 cents per ounce. One ounce of F_1 has 200 units of vitamin A and 120 units of vitamin D, and one ounce of F_2 has 300 units of vitamin A and 40 units of vitamin D.

It is recommended by the Mayo Clinic that every day one should get at least 2700 units per day of vitamin A (this is approximately the mean of the recommended intakes for men and for women) and 200 units per day of vitamin D (for adults up to age 50). How much of each food type should the graduate student purchase to achieve the recommended daily intake of vitamins A and D at the least possible cost?

If x_i is the number of ounces of F_i purchased each day, the daily cost in dollars is $(0.7)x_1 + (0.3)x_2$, and the nutritional value of such a daily purchase is $200x_1 + 300x_2$ for vitamin A and $120x_1 + 40x_2$ for vitamin D. Hence, the goal is to minimize $(0.7)x_1 + (0.3)x_2$ subject to the constraints

$$200x_1 + 300x_2 \geq 2700, \quad 120x_1 + 40x_2 \geq 200, \quad x_1 \geq 0, \quad x_2 \geq 0$$

(the final two inequalities express that the amount purchased should not be negative).

In matrix-vector notation, it is this: for $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbf{R}^2$, minimize $\begin{bmatrix} 7/10 \\ 3/10 \end{bmatrix} \cdot \mathbf{x}$ subject to the constraints

$$\begin{bmatrix} 200 & 300 \\ 120 & 40 \end{bmatrix} \mathbf{x} \geq \begin{bmatrix} 2700 \\ 200 \end{bmatrix},$$

where we use the notational convention for vectors $\mathbf{v}, \mathbf{w} \in \mathbf{R}^n$ that we write $\mathbf{v} \geq \mathbf{w}$ when $v_i \geq w_i$ for all $1 \leq i \leq n$. (We won't use this inequality notation with vectors again, but it is often used elsewhere.) This is a linear programming problem (in the sense of Context II in Section 12.4); expressing such problems in the language of matrices is an essential aspect of efficient solution algorithms for them.

For the combined theme of linear algebra, linear programming, and diets, you may enjoy the Wikipedia page on the Stigler diet and the article [Dan] linked at the bottom there.

■

Example 13.3.12. We can also use matrices to give a shorthand for affine functions. As an example, the affine function $f(x, y, z) = \begin{bmatrix} x - y + z + 3 \\ z - x \\ y + x + 1 \end{bmatrix}$ from Example 13.2.2 can be expressed as

$$f\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix}.$$

Much as inspection of definitions showed that linear functions $\mathbf{R}^n \to \mathbf{R}^m$ are *exactly* those of the form $f(\mathbf{x}) = A\mathbf{x}$ for an $m \times n$ matrix A, affine functions $f : \mathbf{R}^n \to \mathbf{R}^m$ are exactly those of the form $f(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$, where A is an $m \times n$ matrix and $\mathbf{b} \in \mathbf{R}^m$ is a vector. If you think about why this holds for $f : \mathbf{R}^3 \to \mathbf{R}^3$ from Example 13.2.2 that we just considered, the same reasoning works in general.

Affine functions pervade the Kalman filter that filters our noise to ensure accuracy in computer-guided navigation systems as varied as GPS, submarine and aircraft navigation, and moon missions (we'll come back to this in Example 13.5.10). It is also used in econometrics and chemotherapy. For affine functions $A\mathbf{x} + \mathbf{b}$ that arise in this mathematical model, $\mathbf{b}$ corresponds to "random noise".

■

13.4. Further viewpoints on matrix-vector products. Matrix-vector products are a piece of "algebra", but there is a useful *geometric* way to think about their meaning (which we will explore in many ways throughout the rest of the book). The idea is that linear combinations of vectors suggest a certain visualization in our head, and matrix-vector products can be reformulated in that geometric language:

Theorem 13.4.1. If $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n$ are the columns of A (so viewed as vectors in $\mathbf{R}^m$), which is to say

$$A = \begin{bmatrix} | & | & & | \\ \mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_n \\ | & | & & | \end{bmatrix}, \text{ then } A \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1\mathbf{c}_1 + x_2\mathbf{c}_2 + \cdots + x_n\mathbf{c}_n \in \mathbf{R}^m.$$

In particular, the matrix-vector product is a specific linear combination of the columns of the matrix; the vector tells us which linear combination to take (i.e., it records the coefficients $x_1, x_2, \dots, x_n$ in the linear combination).

As we write down the columns $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n$ from *left to right*, each appears in the linear combination multiplied against a scalar that varies through the entries x_i in the "vector" part of the matrix-vector product read from *top to bottom*.

To understand why Theorem 13.4.1 holds, let's look at some examples to see what is going on.

Example 13.4.2. For $g : \mathbf{R}^3 \to \mathbf{R}^3$ as in Example 13.2.2, if we separate the contributions of x, y, and z in its definition we get

$$g\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} x \\ -x \\ x \end{bmatrix} + \begin{bmatrix} -y \\ 0 \\ y \end{bmatrix} + \begin{bmatrix} z \\ z \\ 0 \end{bmatrix} = x\begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} + y\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} + z\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}.$$

On the right side, we are forming linear combinations of vectors using x, y, z as the coefficients, and the vectors in this linear combination are exactly the columns of the 3×3 matrix

$$A = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

that we found in Example 13.3.12 for which $g(\mathbf{v}) = A\mathbf{v}$ for all $\mathbf{v} \in \mathbf{R}^3$. The entries in $\mathbf{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ appear from *top to bottom* as x, y, z, and those are respectively the coefficients of the columns of A from *left to right*: x multiplies the first column, y multiplies the second column, and z multiplies the third column. ■

Example 13.4.3. Returning to the matrix-vector product in Example 13.3.10,

$$\begin{bmatrix} -15 & 5 \\ 55 & -15 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} -15m + 5b \\ 55m - 15b \end{bmatrix} = \begin{bmatrix} -15m \\ 55m \end{bmatrix} + \begin{bmatrix} 5b \\ -15b \end{bmatrix} = m \begin{bmatrix} -15 \\ 55 \end{bmatrix} + b \begin{bmatrix} 5 \\ -15 \end{bmatrix}.$$

Now on the right side we have a linear combination of vectors that are indeed the columns of the given 2×2 matrix, and the coefficients in the linear combination are m and b: the coefficient m of the first column is the top entry in $\begin{bmatrix} m \\ b \end{bmatrix}$ and the coefficient b of the second column is the bottom entry in $\begin{bmatrix} m \\ b \end{bmatrix}$. ■

Example 13.4.4. In the analysis of many electrical networks, for suitable n (depending on the circuit design) the voltage differences and currents at different parts of the circuit can be assembled into n-vectors $\mathbf{V}$ and $\mathbf{I}$ that are related through a matrix-vector product $\mathbf{V} = Z\mathbf{I}$ where Z is an $n \times n$ matrix of "impedance parameters" (whose computation depends on the design of the circuit). ■

For the purpose of stating the next result, we introduce the following useful shorthand:

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbf{R}^n, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \in \mathbf{R}^n, \quad \dots, \quad \mathbf{e}_n = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \in \mathbf{R}^n. \tag{13.4.1}$$

In other words, with n understood from the context, $\mathbf{e}_i$ is the n-vector with a 1 in the ith entry and 0's elsewhere; these are sometimes called *coordinate vectors* or *standard basis vectors* in $\mathbf{R}^n$.

Theorem 13.4.5. For a linear function $\mathbf{f}(\mathbf{x}) = A\mathbf{x}$, the matrix A has as its respective columns $\mathbf{f}(\mathbf{e}_1)$, $\mathbf{f}(\mathbf{e}_2), \dots, \mathbf{f}(\mathbf{e}_n)$, where $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$ are the "coordinate vectors" defined in (13.4.1). In other words:

$$\text{the matrix-vector product } A\mathbf{e}_j \text{ is the } j\text{th column of the matrix } A. \tag{13.4.2}$$

In particular, we can *reconstruct* the matrix A from the function $\mathbf{f}(\mathbf{x}) = A\mathbf{x}$.

The reason (13.4.2) holds in general will become apparent from computations in the examples below.

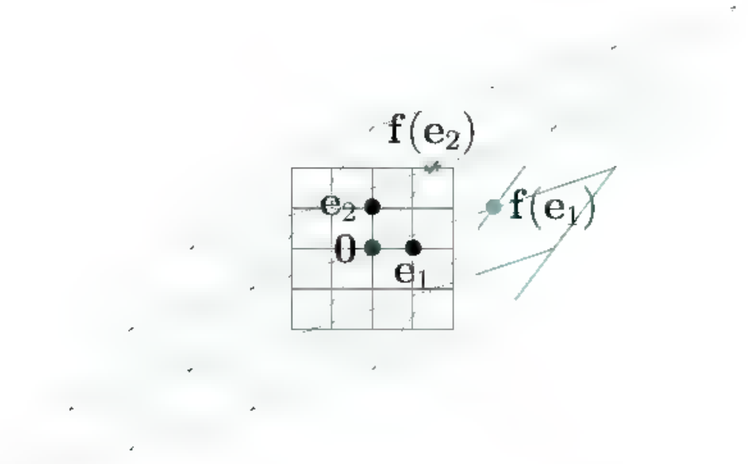
Example 13.4.6. For the matrix

$$A = \begin{bmatrix} 3 & 3/2 \\ 1 & 2 \end{bmatrix},$$

the linear function $\mathbf{f}(\mathbf{x}) = A\mathbf{x}$ satisfies

$$\mathbf{f}(\mathbf{e}_1) = A\mathbf{e}_1 = \begin{bmatrix} 3 & 3/2 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \quad \mathbf{f}(\mathbf{e}_2) = A\mathbf{e}_2 = \begin{bmatrix} 3 & 3/2 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3/2 \\ 2 \end{bmatrix}.$$

Observe that in each case, the matrix-vector product evaluation in the final equality yields exactly the corresponding column of A; this calculation explains why (13.4.2) holds. Figure 13.4.1 shows the effect of $\mathbf{f}$ acting on the usual 2-dimensional unit square grid centered at the origin.



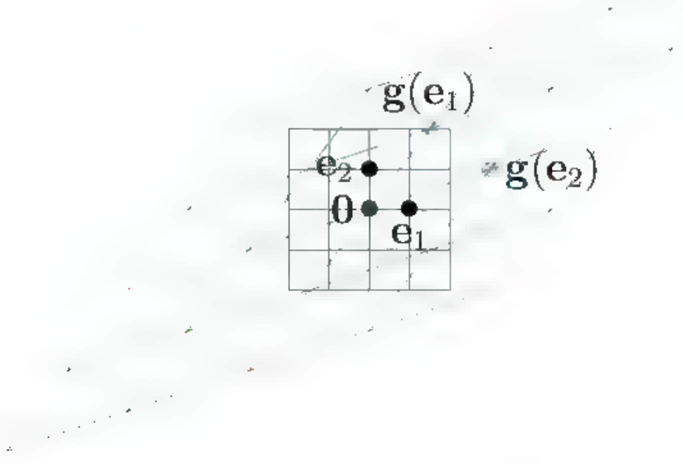
$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} 3 & 3/2 \\ 1 & 2 \end{bmatrix} \mathbf{x}$$

FIGURE 13.4.1. The distorted (blue) grid obtained by applying a linear $\mathbf{f}$ to a (gray) square grid

If we consider the matrix $B = \begin{bmatrix} 3/2 & 3 \\ 2 & 1 \end{bmatrix}$ obtained by *swapping* the columns of A then the associated linear function $\mathbf{g}(\mathbf{x}) = B\mathbf{x}$ is obtained from $\mathbf{f}$ by swapping the placement of the coordinates of $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$ in the output formulas:

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} 3/2 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} (3/2)x + 3y \\ 2x + y \end{bmatrix} = x \begin{bmatrix} 3/2 \\ 2 \end{bmatrix} + y \begin{bmatrix} 3 \\ 1 \end{bmatrix} = x\mathbf{f}(\mathbf{e}_2) + y\mathbf{f}(\mathbf{e}_1)$$

whereas $\mathbf{f}(\mathbf{x}) = \mathbf{f}(x\mathbf{e}_1 + y\mathbf{e}_2) = x\mathbf{f}(\mathbf{e}_1) + y\mathbf{f}(\mathbf{e}_2)$. In particular, $\mathbf{g}(\mathbf{e}_1) = \mathbf{f}(\mathbf{e}_2)$ and $\mathbf{g}(\mathbf{e}_2) = \mathbf{f}(\mathbf{e}_1)$. Figure 13.4.2 shows the effect of $\mathbf{g}$ on the unit square grid centered at the origin.



$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} 3/2 & 3 \\ 2 & 1 \end{bmatrix} \mathbf{x}$$

FIGURE 13.4.2. The function $\mathbf{g}$ yields the same blue grid, with roles of $\mathbf{f}(\mathbf{e}_1)$ and $\mathbf{f}(\mathbf{e}_2)$ swapped

■

Example 13.4.7. Now consider the matrix

$$C = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/2 & 1/2 & 1/\sqrt{2} \\ 1/2 & -1/2 & 1/\sqrt{2} \end{bmatrix}$$

and its associated linear function $\mathbf{f} : \mathbf{R}^3 \to \mathbf{R}^3$ defined by $\mathbf{f}(\mathbf{x}) = C\mathbf{x}$. In Section 14.5 we will see how C arises in the context of a natural geometric question, but for now we'd like to see that the link between the columns of C and the outputs $\mathbf{f}(\mathbf{e}_j)$ gives us a vivid way to visualize the effect of $\mathbf{f}$. By direct computation we see that

$$\mathbf{f}(\mathbf{e}_1) = C\mathbf{e}_1 = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/2 & 1/2 & 1/\sqrt{2} \\ 1/2 & -1/2 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} \\ -1/2 \\ 1/2 \end{bmatrix}$$

(check the final equality for yourself by hand); this is exactly the first column of C, in accordance with (13.4.2). Similarly, one checks that

$$\mathbf{f}(\mathbf{e}_2) = C\mathbf{e}_2 = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/2 & 1/2 & 1/\sqrt{2} \\ 1/2 & -1/2 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} \\ 1/2 \\ -1/2 \end{bmatrix}$$

and

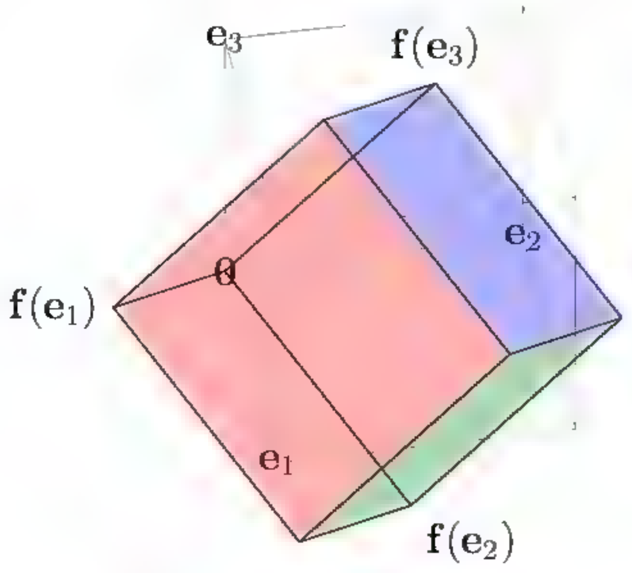
$$\mathbf{f}(\mathbf{e}_3) = C\mathbf{e}_3 = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/2 & 1/2 & 1/\sqrt{2} \\ 1/2 & -1/2 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix},$$

again affirming (13.4.2). By checking the final equalities for each of these computations of $\mathbf{f}(\mathbf{e}_j)$ (with $j = 1, 2, 3$), you can see the pattern which makes (13.4.2) hold in general.

The interesting feature of this example is that *the columns of C are pairwise orthogonal unit vectors*. This property of the columns is verified by a direct calculation: each column has length 1 and dot products of distinct columns all vanish (compute for yourself). With this in hand, it follows that

the vectors $\mathbf{f}(\mathbf{e}_1), \mathbf{f}(\mathbf{e}_2), \mathbf{f}(\mathbf{e}_3)$ are pairwise orthogonal unit vectors

since the $\mathbf{f}(\mathbf{e}_j)$'s are exactly the columns of C. In Section 20.4 we will see that this implies that $\mathbf{f}$ *preserves lengths and angles, so it is a rigid motion of space*. We can thereby visualize the effect of $\mathbf{f}$ in terms of carrying a "unit cube" with edges $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ to a "unit cube" with edges $\mathbf{f}(\mathbf{e}_1), \mathbf{f}(\mathbf{e}_2), \mathbf{f}(\mathbf{e}_3)$.


$\mathbf{e}_3$
$\mathbf{f}(\mathbf{e}_3)$
$\mathbf{e}_2$
$\mathbf{f}(\mathbf{e}_1)$
0
$\mathbf{e}_1$
$\mathbf{f}(\mathbf{e}_2)$

■

You should memorize the formulas in Definition 13.3.4, Theorem 13.4.1, and Theorem 13.4.5 (hopefully made more feasible by working through the preceding examples). Matrix-vector products are just a shorthand notation for a certain type of operation on vectors, but all work in linear algebra relies on it.

13.5. The derivative matrix. Now let us rewrite (13.1.2) using matrix shorthand:

$$\mathbf{f}(x,y,z) \approx \mathbf{f}(a,b,c) + \begin{bmatrix} \frac{\partial f_1}{\partial x}(a,b,c) & \frac{\partial f_1}{\partial y}(a,b,c) & \frac{\partial f_1}{\partial z}(a,b,c) \\ \frac{\partial f_2}{\partial x}(a,b,c) & \frac{\partial f_2}{\partial y}(a,b,c) & \frac{\partial f_2}{\partial z}(a,b,c) \\ \frac{\partial f_3}{\partial x}(a,b,c) & \frac{\partial f_3}{\partial y}(a,b,c) & \frac{\partial f_3}{\partial z}(a,b,c) \end{bmatrix} \begin{bmatrix} x-a \\ y-b \\ z-c \end{bmatrix}.$$

If we use the more efficient vector notation $\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ and $\mathbf{a} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ then this can be written as:

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{f}(\mathbf{a}) + \begin{bmatrix} \frac{\partial f_1}{\partial x}(\mathbf{a}) & \frac{\partial f_1}{\partial y}(\mathbf{a}) & \frac{\partial f_1}{\partial z}(\mathbf{a}) \\ \frac{\partial f_2}{\partial x}(\mathbf{a}) & \frac{\partial f_2}{\partial y}(\mathbf{a}) & \frac{\partial f_2}{\partial z}(\mathbf{a}) \\ \frac{\partial f_3}{\partial x}(\mathbf{a}) & \frac{\partial f_3}{\partial y}(\mathbf{a}) & \frac{\partial f_3}{\partial z}(\mathbf{a}) \end{bmatrix} (\mathbf{x} - \mathbf{a}). \tag{13.5.1}$$

Example 13.3.7 illustrated this for $\mathbf{g}$ from Example 13.1.1: for (x, y, z) near $(2, 1, 0)$ we have

$$\begin{bmatrix} e^z(x-y)^2 \\ 2yz + x^3 \\ x^2 - y^3 + z \end{bmatrix} \approx \begin{bmatrix} 1 \\ 8 \\ 3 \end{bmatrix} + \begin{bmatrix} 2 & -2 & 1 \\ 12 & 0 & 2 \\ 4 & -3 & 1 \end{bmatrix} \begin{bmatrix} x-2 \\ y-1 \\ z \end{bmatrix}.$$

Let's understand what is going on in (13.5.1) from a broader perspective. For each component function f_i of $\mathbf{f}$ we have the linear approximation $f_i(\mathbf{x}) \approx f_i(\mathbf{a}) + ((\nabla f_i)(\mathbf{a})) \cdot (\mathbf{x} - \mathbf{a})$ for $\mathbf{x}$ near $\mathbf{a}$. Assemble these scalar approximations into an approximation to the vector $\mathbf{f}(\mathbf{x})$. The scalars $f_i(\mathbf{x})$ and $f_i(\mathbf{a})$ are the respective entries in the vectors $\mathbf{f}(\mathbf{x})$ and $\mathbf{f}(\mathbf{a})$, and (in accordance with the *definition* of matrix-vector products in Definition 13.3.4) the dot products $((\nabla f_i)(\mathbf{a})) \cdot (\mathbf{x} - \mathbf{a})$ are the entries in $A(\mathbf{x} - \mathbf{a})$ for the 3×3 matrix A with ith *row* $(\nabla f_i)(\mathbf{a})$. (More generally, if $\mathbf{v}$ is an n-vector then the function $\mathbf{R}^n \to \mathbf{R}$ defined by $T(\mathbf{x}) = \mathbf{v} \cdot \mathbf{x} = \sum v_i x_i$ is $M\mathbf{x}$ for the $1 \times n$ "row" matrix M with ith entry v_i.)

We repeat for emphasis: for $\mathbf{x}$ near $\mathbf{a}$, when approximating $\mathbf{f}(\mathbf{x})$ by $\mathbf{f}(\mathbf{a}) + A(\mathbf{x} - \mathbf{a})$ for a 3×3 matrix A, the *definition* of matrix-vector products makes the gradients at $\mathbf{a}$ of the components f_i naturally appear as the *rows* (not as the columns!) of A, as in (13.5.1). This leads us to a general concept:

Definition 13.5.1. Let $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ be a vector-valued function $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix}$ with scalar-valued components $f_1, \dots, f_m : \mathbf{R}^n \to \mathbf{R}$. The *derivative matrix* of $\mathbf{f}$ at a point $\mathbf{a} \in \mathbf{R}^n$ is the $m \times n$ matrix

$$(D\mathbf{f})(\mathbf{a}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{a}) & \frac{\partial f_1}{\partial x_2}(\mathbf{a}) & \dots & \frac{\partial f_1}{\partial x_n}(\mathbf{a}) \\ \frac{\partial f_2}{\partial x_1}(\mathbf{a}) & \frac{\partial f_2}{\partial x_2}(\mathbf{a}) & \dots & \frac{\partial f_2}{\partial x_n}(\mathbf{a}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(\mathbf{a}) & \frac{\partial f_m}{\partial x_2}(\mathbf{a}) & \dots & \frac{\partial f_m}{\partial x_n}(\mathbf{a}) \end{bmatrix}$$

with all partial derivatives $\partial f_i / \partial x_j$ evaluated at the point $\mathbf{a}$. (In other books or courses you may see this denoted as $D\mathbf{f}(\mathbf{a})$ or called the "Jacobian matrix"; we will not use that notation and terminology.)

Warning: For a scalar-valued $f : \mathbf{R}^n \to \mathbf{R}$ $(m = 1)$, the matrix $(Df)(\mathbf{a})$ is a "flipped" gradient:

$$(Df)(\mathbf{a}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\mathbf{a}) & \frac{\partial f}{\partial x_2}(\mathbf{a}) & \dots & \frac{\partial f}{\partial x_n}(\mathbf{a}) \end{bmatrix}, \text{ whereas } (\nabla f)(\mathbf{a}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\mathbf{a}) \\ \frac{\partial f}{\partial x_2}(\mathbf{a}) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\mathbf{a}) \end{bmatrix}.$$

In general (any m), the ith row of $(D\mathbf{f})(\mathbf{a})$ is "$(\nabla f_i)(\mathbf{a})$ written horizontally"; i.e., $(D\mathbf{f})(\mathbf{a})$ is an $m \times n$ matrix whose ith row encodes the n-vector gradient at $\mathbf{a}$ of the ith component function f_i of $\mathbf{f}$. If you are bothered by this "flipping", note that it appeared naturally in (13.5.1) as well as in the discussion immediately after that as a consequence of how matrix-vector products are *defined*.

Example 13.5.2. For $\mathbf{g} : \mathbf{R}^3 \to \mathbf{R}^3$ as in Example 13.1.1 and $\mathbf{a} = (2, 1, 0)$ we have

$$(D\mathbf{g})(\mathbf{a}) = \begin{bmatrix} 2 & -2 & 1 \\ 12 & 0 & 2 \\ 4 & -3 & 1 \end{bmatrix}.$$

For instance, the component function $g_1(x, y, z) = e^z(x - y)^2$ has gradient at $\mathbf{a} = (2, 1, 0)$ equal to $(\nabla g_1)(\mathbf{a}) = \begin{bmatrix} 2 \\ -2 \\ 1 \end{bmatrix}$, and correspondingly the first row of $(D\mathbf{g})(\mathbf{a})$ is the "flipped" version of this gradient vector. ■

Example 13.5.3. Let $\mathbf{f} : \mathbf{R}^3 \to \mathbf{R}^2$ be defined by $\mathbf{f}(x, y, z) = (x^2 - y, z^3 + xy)$. Write $f_1(x, y, z) = x^2 - y$ and $f_2(x, y, z) = z^3 + xy$. Then the derivative matrix is the 2×3 matrix

$$(D\mathbf{f})(x, y, z) = \begin{bmatrix} \partial f_1/\partial x & \partial f_1/\partial y & \partial f_1/\partial z \\ \partial f_2/\partial x & \partial f_2/\partial y & \partial f_2/\partial z \end{bmatrix} = \begin{bmatrix} 2x & -1 & 0 \\ y & x & 3z^2 \end{bmatrix}.$$

For example, the value of the derivative matrix at the point $(1, 1, 1)$ is the matrix $\begin{bmatrix} 2 & -1 & 0 \\ 1 & 1 & 3 \end{bmatrix}$. ■

Example 13.5.4. Let $\mathbf{f} : \mathbf{R}^2 \to \mathbf{R}^3$ be defined by $\mathbf{f}(x_1, x_2) = (x_2^3 - 4x_1, x_2 e^{\ x_1}, \cos(x_1 x_2))$. Write $f_1(x_1, x_2) = x_2^3 - 4x_1$, $f_2(x_1, x_2) = x_2 e^{\ x_1}$, and $f_3(x_1, x_2) = \cos(x_1 x_2)$. Then the derivative matrix is the 3×2 matrix

$$(D\mathbf{f})(x_1, x_2) = \begin{bmatrix} \partial f_1/\partial x_1 & \partial f_1/\partial x_2 \\ \partial f_2/\partial x_1 & \partial f_2/\partial x_2 \\ \partial f_3/\partial x_1 & \partial f_3/\partial x_2 \end{bmatrix} = \begin{bmatrix} -4 & 3x_2^2 \\ -x_2 e^{\ x_1} & e^{\ x_1} \\ -x_2 \sin(x_1 x_2) & -x_1 \sin(x_1 x_2) \end{bmatrix}.$$ ■

Example 13.5.5. Let $\mathbf{f} : \mathbf{R}^2 \to \mathbf{R}^2$ be defined by $\mathbf{f}(x, y) = (x^3 + xy + 2y^3 - 3, x^2 y + xy^2 - y)$. Write $f_1(x, y) = x^3 + xy + 2y^3 - 3$ and $f_2(x, y) = x^2 y + xy^2 - y$. Then the derivative matrix is the 2×2 matrix

$$(D\mathbf{f})(x, y) = \begin{bmatrix} \partial f_1/\partial x & \partial f_1/\partial y \\ \partial f_2/\partial x & \partial f_2/\partial y \end{bmatrix} = \begin{bmatrix} 3x^2 + y & x + 6y^2 \\ 2xy + y^2 & x^2 + 2xy - 1 \end{bmatrix}.$$

We will come back to this $\mathbf{f}$ in Example 18.5.6. ■

Example 13.5.6. For a function $f : \mathbf{R}^n \to \mathbf{R}$, its gradient is a function $\mathbf{g} = \nabla f : \mathbf{R}^n \to \mathbf{R}^n$. For $\mathbf{a} \in \mathbf{R}^n$, let's work out the entries in the $n \times n$ matrix $(D\mathbf{g})(\mathbf{a})$ in terms of f. By definition of $\mathbf{g} = \nabla f$, the ij-entry is $(\partial g_i/\partial x_j)(\mathbf{a})$ with $g_i = \partial f/\partial x_i$ the ith component function of $\mathbf{g}$.

In other words, the ij-entry is $((\partial/\partial x_j)(\partial f/\partial x_i))(\mathbf{a}) = (\partial^2 f/\partial x_j \partial x_i)(\mathbf{a})$. Thus, by equality of mixed partials (Theorem 9.6.4) the ij-entry and the ji-entry of $(D(\nabla f))(\mathbf{a})$ *coincide*. That property of a square matrix (equality of the ij-entry and ji-entry for all i, j, to be called "symmetric") will turn out to have remarkable consequences (discussed in Section 24.1). If $\mathbf{a}$ is a critical point of f then the $n \times n$ matrix $(D(\nabla f))(\mathbf{a}) = ((\partial^2 f/\partial x_i \partial x_j)(\mathbf{a}))$ of second partial derivatives of f at $\mathbf{a}$ holds the key to the multivariable second derivative test at $\mathbf{a}$, as we will see in Chapter 26.

As a sample calculation, for $F(x, y, z) = x^2 yz + \cos(xy) + ze^y$ let's compute $(D(\nabla F))(\mathbf{a})$ with $\mathbf{a} = (-1, 0, 3)$. To do this, we will compute the first and then second partial derivatives symbolically, and finally evaluate at the point of interest. The first partial derivatives are

$$F_x = 2xyz - y\sin(xy), \quad F_y = x^2 z - x\sin(xy) + ze^y, \quad F_z = x^2 y + e^y,$$

and from these we compute the second partial derivatives:

$$\partial^2 F/\partial x^2 = 2yz - y^2\cos(xy),\ \ \partial^2 F/\partial y^2 = -x^2\cos(xy) + ze^y,\ \ \partial^2 F/\partial z^2 = 0,$$
$$\partial^2 F/\partial x\partial y = \partial^2 F/\partial y\partial x = 2xz - \sin(xy) - xy\cos(xy),\ \ \partial^2 F/\partial x\partial z = \partial^2 F/\partial z\partial x = 2xy,$$
$$\partial^2 F/\partial y\partial z = \partial^2 F/\partial z\partial y = x^2 + e^y.$$

Accordingly, $(D(\nabla F))(x,y,z)$ is equal to

$$\begin{bmatrix} 2yz - y^2\cos(xy) & 2xz - \sin(xy) - xy\cos(xy) & 2xy \\ 2xz - \sin(xy) - xy\cos(xy) & -x^2\cos(xy) + ze^y & x^2 + e^y \\ 2xy & x^2 + e^y & 0 \end{bmatrix}, \tag{13.5.2}$$

so

$$(D(\nabla F))(-1,0,3) = \begin{bmatrix} 0 & -6 & 0 \\ -6 & 2 & 2 \\ 0 & 2 & 0 \end{bmatrix}.$$

By inspection, indeed the ij-entry and ji-entry coincide for any i,j (e.g., the $(1,2)$-entry and $(2,1)$-entry both equal -6). ■

Example 13.5.7. Derivative matrices are a central tool in the study of deformations of solid bodies. For a physical body B subjected to a deformation, the applied forces constitute the "stress" and the resulting change in its shape is the "strain". Contexts where such situations arise include the behavior of components of the Golden Gate Bridge due to the wind and motion of vehicles, the shaking of the ground during an earthquake (both pressure waves and shear waves), squeezing a piece of rubber, compression springs in cars, changes in the optical properties of a crystal under pressure, and stuffing a piece of cork into the top of a wine bottle.

Define $\mathbf{f} : B \to \mathbf{R}^3$ by letting $\mathbf{f}(\mathbf{b})$ be the point in space where a point $\mathbf{b} \in B$ winds up after the deformation process. Engineers call the 3×3 matrix $(D\mathbf{f})(\mathbf{b})$ for $\mathbf{b} \in B$ the *deformation gradient* at $\mathbf{b}$ (and it has useful dynamical analogues in fluid mechanics by replacing $\mathbf{f}$ with a vector field expressing the trajectory of fluid flow at all points at a given time). This use of the word "gradient" is unfortunate since it is not a vector but rather is a matrix; the terminology "deformation matrix" is also used, but is much less common.

The deformation $\mathbf{f}$ is called *homogeneous* when it is affine: $\mathbf{f}(\mathbf{b}) = M\mathbf{b} + \mathbf{v}$ for some $\mathbf{v} \in \mathbf{R}^3$ and 3×3 matrix M that are independent of $\mathbf{b}$ (a 2-dimensional visualization with $\mathbf{v} = \mathbf{0}$ is given in Figure 14.1.5 with the unit square S of points (x,y) with $0 \leq x,y \leq 1$ in the role of B). The function $\mathbf{g}(\mathbf{x}) = M\mathbf{x} + \mathbf{v}$ has derivative matrix $(D\mathbf{g})(\mathbf{a}) = M$ for all $\mathbf{a}$, generalizing that functions of the form $g(x) = mx + c$ in single-variable calculus have constant derivative: $g'(a) = m$ for all a. Homogeneous deformations thereby correspond to the case that $(D\mathbf{f})(\mathbf{b})$ is independent of $\mathbf{b}$. The utility of the deformation gradient rests on techniques from "matrix algebra", as we will discuss later (in Example 24.6.7). ■

The reason for the importance of the derivative matrix is that it encodes the idea of "best linear approximation" generalizing the "tangent line" interpretation of derivatives in single-variable calculus. To explain this, recall that in single-variable calculus, for a function $f : \mathbf{R} \to \mathbf{R}$ the derivative $f'(a)$ at a point $a \in \mathbf{R}$ is a number that gives the slope of the tangent line to the graph of $y = f(x)$ at the point $(a, f(a))$. In algebraic terms, approximating a tangent line by a secant line and then regarding the slope of the latter as an approximation to the slope of the former can be written as

$$f'(a) \approx \frac{f(x) - f(a)}{x - a}$$

for x near a. (This encodes the limit definition of $f'(a)$.) Written this way, there is no meaningful analogue when the number $x \in \mathbf{R}$ is replaced by a vector $\mathbf{x} \in \mathbf{R}^n$ for $n > 1$ (it makes no sense to "divide by $\mathbf{x}-\mathbf{a}$").

But we can rewrite the approximation in another way that avoids division: for x near a we have $f(x) \approx f(a) + f'(a)(x-a)$, or equivalently for h near 0 we have

$$f(a+h) \approx f(a) + f'(a)h$$

(the best way to think about the meaning of $f'(a)$!). The right side arises in the equation of the tangent line to the graph at $x = a$: $y = f(a) + f'(a)(x-a)$. In this latter formulation, the special property of the number $f'(a)$ is that the number $c \in \mathbf{R}$ for which the line $y = f(a) + c(x-a)$ best approximates the graph to $y = f(x)$ for x near a is $c = f'(a)$. This brings us to:

Informal Question. For $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ and a point $\mathbf{a} \in \mathbf{R}^n$, is there a linear function $L : \mathbf{R}^n \to \mathbf{R}^m$ (depending on $\mathbf{a}$) so that

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{f}(\mathbf{a}) + L(\mathbf{x}-\mathbf{a})$$

for $\mathbf{x}$ near $\mathbf{a}$? Or equivalently,

$$\mathbf{f}(\mathbf{a}+\mathbf{h}) \approx \mathbf{f}(\mathbf{a}) + L(\mathbf{h})$$

for n-vectors $\mathbf{h}$ near $\mathbf{0}$?

If there is some L (depending on $\mathbf{a}$) that makes the approximation "best possible" in the sense that the size of the error decays to 0 more rapidly than $\|\mathbf{x}-\mathbf{a}\|$ as $\mathbf{x}$ approaches $\mathbf{a}$ (or more rapidly than $\|\mathbf{h}\|$ as $\mathbf{h}$ approaches $\mathbf{0}$) then L is unique and we call it the *best linear approximation* to $\mathbf{f}$ at $\mathbf{a}$.

The derivative matrix is the affirmative answer to this question, for reasons we saw immediately before Definition 13.5.1 (namely, that its rows are the gradients of the component functions of $\mathbf{f}$):

Theorem 13.5.8. The best linear approximation to $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ at $\mathbf{a} \in \mathbf{R}^n$ is given by the $m \times n$ derivative matrix $(D\mathbf{f})(\mathbf{a})$: we have the optimal approximation of m-vectors

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{f}(\mathbf{a}) + \underbrace{((D\mathbf{f})(\mathbf{a}))(\mathbf{x}-\mathbf{a})}_{\text{matrix-vector multiplication}} \tag{13.5.3}$$

for n-vectors $\mathbf{x}$ near $\mathbf{a}$, or equivalently

$$\mathbf{f}(\mathbf{a}+\mathbf{h}) \approx \mathbf{f}(\mathbf{a}) + \underbrace{((D\mathbf{f})(\mathbf{a}))\,\mathbf{h}}_{\text{matrix-vector multiplication}} \tag{13.5.4}$$

for n-vectors $\mathbf{h}$ near $\mathbf{0}$.

Example 13.5.9. For $\mathbf{f} : \mathbf{R}^3 \to \mathbf{R}^2$ defined by $\mathbf{f}(x,y,z) = (x^2 - y, z^3 + xy)$, let's work out its best linear approximations in the senses of (13.5.3) and (13.5.4) at the point $(1,1,1)$. We computed the 2×3 matrix $(D\mathbf{f})(x,y,z)$ symbolically in Example 13.5.3, from which we obtained

$$(D\mathbf{f})(1,1,1) = \begin{bmatrix} 2 & -1 & 0 \\ 1 & 1 & 3 \end{bmatrix}.$$

Hence, for (x,y,z) near $(1,1,1)$ we have

$$\begin{aligned} \mathbf{f}(x,y,z) \approx \mathbf{f}(1,1,1) + \begin{bmatrix} 2 & -1 & 0 \\ 1 & 1 & 3 \end{bmatrix} \begin{bmatrix} x-1 \\ y-1 \\ z-1 \end{bmatrix} &= \begin{bmatrix} 0 \\ 2 \end{bmatrix} + \begin{bmatrix} 2(x-1)-(y-1) \\ (x-1)+(y-1)+3(z-1) \end{bmatrix} \\ &= \begin{bmatrix} -1+2x-y \\ -3+x+y+3z \end{bmatrix} \end{aligned}$$

and for (h_1, h_2, h_3) near $\mathbf{0}$ we have

$$\mathbf{f}(1+h_1, 1+h_2, 1+h_3) \approx \mathbf{f}(1,1,1) + \begin{bmatrix} 2 & -1 & 0 \\ 1 & 1 & 3 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \end{bmatrix} + \begin{bmatrix} 2h_1 - h_2 \\ h_1 + h_2 + 3h_3 \end{bmatrix}$$

$$= \begin{bmatrix} 2h_1 - h_2 \\ 2 + h_1 + h_2 + 3h_3 \end{bmatrix}.$$

In other words, for (x, y, z) near $(1, 1, 1)$ we have $(x^2 - y, z^3 + xy) \approx (-1 + 2x - y, -3 + x + y + 3z)$ and for (h_1, h_2, h_3) near $\mathbf{0}$ we have $\mathbf{f}(1 + h_1, 1 + h_2, 1 + h_3) \approx (2h_1 - h_2, 2 + h_1 + h_2 + 3h_3)$. ■

The role of the matrix $(D\mathbf{f})(\mathbf{a})$ in expressing a "best linear approximation" for $\mathbf{f}$ near $\mathbf{a}$ is the key to unlocking the powerful role of linear algebra in the study of general non-linear $\mathbf{f}$ in multivariable calculus. In order to carry this out, we need to gain more experience with matrices and their role in linear algebra (much as you spent a fair amount of time learning how to graph straight lines before you learned the derivative aspects of calculus). The next few chapters are devoted to this task.

Example 13.5.10. At the end of Example 13.3.12 we mentioned that functions of the form $\mathbf{f}(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$ arise in an important statistical model (called the Kalman filter) for managing noise in navigation systems. But for many applications, from the Apollo moon missions to GPS and beyond, it is essential to incorporate *non-linear* phenomena and so the original Kalman filter cannot be applied. Approximating a non-linear function by a linear one via derivative matrices leads to a useful alternative statistical model (called the extended Kalman filter) for non-linear settings. ■

Example 13.5.11. Derivative matrices show up everywhere in robotics (where they are usually called by the name "Jacobian"). Consider a robotic system that moves within a plane: it consists of an arm of length L_1 that is fixed at one end to a point P (around which it rotates freely) and a second arm of length L_2 that is attached to the first arm at its other endpoint Q (at which the second arm rotates freely). This has 2 degrees of freedom (corresponding to angles measured at P and Q), and we are interested in the position (x, y) of the other endpoint R of the second arm (where it is not attached to the first arm); this is the tip of the robot, where there may be a paintbrush, blade, etc.

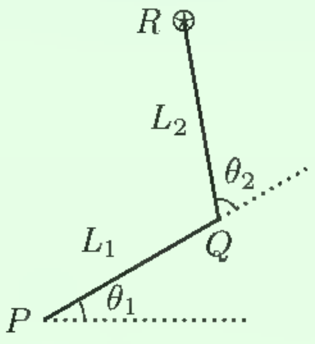


FIGURE 13.5.1. A robotic system based at P with "elbow" at Q and tip at R

To describe the physical configuration at a given time, we set up a coordinate system in the plane of motion with origin at P and measure two angles as shown in Figure 13.5.1: the angle θ_1 that the first arm makes at P relative to the x-axis, and the angle θ_2 that the second arm makes at Q relative to the line through the first arm. Since the second arm makes an angle $\theta_1 + \theta_2$ relative to the x-axis, the point Q is located at $(L_1 \cos(\theta_1), L_1 \sin(\theta_1))$ and the displacement from Q to R is

$(L_2\cos(\theta_1+\theta_2), L_2\sin(\theta_1+\theta_2))$. Hence, as a function of the angles, the position of the tip R of the robotic arm is given by

$$\mathbf{p}(\theta_1,\theta_2) = \begin{bmatrix} x(\theta_1,\theta_2) \\ y(\theta_1,\theta_2) \end{bmatrix} = \begin{bmatrix} L_1\cos(\theta_1) + L_2\cos(\theta_1+\theta_2) \\ L_1\sin(\theta_1) + L_2\sin(\theta_1+\theta_2) \end{bmatrix}. \tag{13.5.5}$$

The linear approximation property of derivative matrices, applied to $\mathbf{p}(\theta_1,\theta_2)$, is used to compute the approximate small change $\Delta\vec{\theta}$ in the 2-vector $\vec{\theta}$ of angles achieving a desired small change $\Delta\mathbf{p}$ in the position of the tip R. We will work this out in Example 18.5.7 using more knowledge of matrices, and apply it to compute the configurations where the robot gets stuck or behaves unpredictably. ■

Remark 13.5.12. The approximations on the right sides of (13.5.3) and (13.5.4) are *affine* functions of $\mathbf{x}-\mathbf{a}$ and $\mathbf{h}$ respectively, due to the addition of the vector $\mathbf{f}(\mathbf{a})$ that is usually nonzero. Nonetheless, everyone refers to (13.5.3) and (13.5.4) as the "best linear approximation" (even though as a functions of $\mathbf{x}-\mathbf{a}$ and $\mathbf{h}$ they are typically just affine rather than linear). This informal terminology matches what we say in single-variable calculus, calling the tangent-line expression $f(a)+f'(a)(x-a)$ for the graph of $f:\mathbf{R}\to\mathbf{R}$ at the point $(a, f(a))$ the "best linear approximation" to f at $x=a$.

For a function $h:\mathbf{R}^2\to\mathbf{R}$, the associated affine function $\mathbf{R}^2\to\mathbf{R}$ as on the right side of (13.5.3) for the function h and $\mathbf{a}=(a,b)\in\mathbf{R}^2$ is the approximation as in (11.1.3) that appears in an equation for the tangent plane to the surface graph $z=h(x,y)$ at $(a,b,h(a,b))$ (see (11.2.3)). Figures 11.1.2 and 11.2.3 illustrate such a tangent plane as an approximation to the surface graph at the chosen point $\mathbf{a}$. This is a 2-dimensional analogue of the tangent line approximation to a graph in single-variable calculus.

Remark 13.5.13. Strictly speaking, Theorem 13.5.8 requires the assumption that $\mathbf{f}:\mathbf{R}^n\to\mathbf{R}^m$ is "differentiable", so it gets involved with an issue swept under the rug in Remark 10.2.4 when $m=1$: we have not defined what "differentiable" means for a function of n variables when $n>1$. We did however mention there that for $f:\mathbf{R}^n\to\mathbf{R}$, this concept (whatever it is to mean) is a consequence of the existence and continuity of all partial derivatives $\partial f/\partial x_j:\mathbf{R}^n\to\mathbf{R}$; the same holds for $\mathbf{f}:\mathbf{R}^n\to\mathbf{R}^m$ when all $\partial f_i/\partial x_j:\mathbf{R}^n\to\mathbf{R}$ exist and are continuous (so in practice it is fulfilled).

In fact, Theorem 13.5.8 hints at the actual *definition* of differentiability for $\mathbf{f}:\mathbf{R}^n\to\mathbf{R}^m$ at a point $\mathbf{a}\in\mathbf{R}^n$ for any n: the definition is that there should exist an $m\times n$ matrix $L_{\mathbf{a}}$ (depending on $\mathbf{a}$) for which $\mathbf{f}(\mathbf{a}+\mathbf{h})\approx\mathbf{f}(\mathbf{a})+L_{\mathbf{a}}\mathbf{h}$ for all small $\mathbf{h}$, with "$\approx$" encoding that the error in the approximation is "of smaller order than $\|\mathbf{h}\|$ as $\mathbf{h}$ approaches $\mathbf{0}$" (so for all small $\mathbf{h}$ the deviation of $\mathbf{f}(\mathbf{a}+\mathbf{h})$ from $\mathbf{f}(\mathbf{a})$ is well-approximated by the function of $\mathbf{h}$ given by the matrix-vector product $L_{\mathbf{a}}\mathbf{h}$).

More precisely, the *definition* of differentiability of $\mathbf{f}:\mathbf{R}^n\to\mathbf{R}^m$ at $\mathbf{a}\in\mathbf{R}^n$ is that there is an $m\times n$ matrix $L_{\mathbf{a}}$ so that as $\mathbf{h}$ approaches $\mathbf{0}$ we have $\dfrac{\|\mathbf{f}(\mathbf{a}+\mathbf{h})-(\mathbf{f}(\mathbf{a})+L_{\mathbf{a}}\mathbf{h})\|}{\|\mathbf{h}\|}\to 0$ (where the left side is a ratio of *numbers*). Note that when $m=n=1$ we recover the limit definition of differentiability in single-variable calculus with L_a given by the 1×1 matrix $[f'(a)]$. For general m and n it is a **theorem** that when such an $L_{\mathbf{a}}$ exists for a given $\mathbf{a}$ then *necessarily* the partial derivatives $\partial f_i/\partial x_j$ exist at $\mathbf{a}$ (so $(D\mathbf{f})(\mathbf{a})$ makes sense) and moreover $L_{\mathbf{a}}=(D\mathbf{f})(\mathbf{a})$.

That $(D\mathbf{f})(\mathbf{a})$ is the "best linear approximation" to $\mathbf{f}$ at $\mathbf{a}$ when $\mathbf{f}$ is differentiable is not baked into the definitions, but rather also has to be proved (i.e., Theorem 13.5.8 has not become an empty assertion, though its proof turns out not to be difficult). A more difficult result to establish is that existence and continuity of partial derivatives of the component functions f_i ensures differentiability in the sense just defined.

Chapter 13 highlights (links to highlights in previous and next chapters)

Notation	**Meaning**	**Location in text**
$a_{ij}, a_{i,j}$	entry in ith row from top and jth column from left in matrix A	Definition 13.3.1
$A\mathbf{x}$	matrix-vector product	Definition 13.3.4
$(D\mathbf{f})(\mathbf{a})$	$m \times n$ derivative matrix at $\mathbf{a} \in \mathbf{R}^n$ for $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$	Definition 13.5.1

Concept	**Meaning**	**Location in text**
linear function $\mathbf{R}^n \to \mathbf{R}^m$	function $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ whose component functions $f_i : \mathbf{R}^n \to \mathbf{R}$ have the form $a_{i1}x_1 + \cdots + a_{in}x_n$	Definition 13.2.1
affine function $\mathbf{R}^n \to \mathbf{R}^m$	function $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ whose component functions $f_i : \mathbf{R}^n \to \mathbf{R}$ have the form $a_{i1}x_1 + \cdots + a_{in}x_n + b_i$	Definition 13.2.1
$m \times n$ matrix	array of m horizontal rows and n vertical columns	Def. 13.3.1, Ex. 13.3.2
ij-entry, (i, j)-entry	entry in ith row from top and jth column from left in a matrix	Definition 13.3.1
matrix-vector product	for $m \times n$ matrix A and n-vector $\mathbf{x}$, it is the m-vector with ith entry $\sum_{j=1}^{n} a_{ij}x_j$	Definition 13.3.4
derivative matrix	for $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ and $\mathbf{a} \in \mathbf{R}^n$, $(D\mathbf{f})(\mathbf{a})$ has ij-entry $(\partial f_i/\partial x_j)(\mathbf{a})$	Definition 13.5.1
best linear approximation	for $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ and $\mathbf{a} \in \mathbf{R}^n$, the affine function $\mathbf{R}^n \to \mathbf{R}^m$ carrying $\mathbf{x}$ to $\mathbf{f}(\mathbf{a}) + ((D\mathbf{f})(\mathbf{a}))(\mathbf{x} - \mathbf{a})$	Theorem 13.5.8

Result	**Meaning**	**Location in text**
linear functions via matrix-vector products	linear functions $\mathbf{R}^n \to \mathbf{R}^m$ are precisely those of the form $\mathbf{f}(\mathbf{x}) = A\mathbf{x}$ for an $m \times n$ matrix A (so matrices are exactly a means of encoding linear functions; matrices as arrays of scalars arise in many contexts, but the link to linear functions will be the key to their utility)	Proposition 13.3.8
affine functions via matrix-vector products	affine functions $\mathbf{R}^n \to \mathbf{R}^m$ are precisely those of the form $\mathbf{f}(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$ for an $m \times n$ matrix A and m-vector $\mathbf{b}$	end of Ex. 13.3.12
$A\mathbf{x}$ is linear combination of columns	for $m \times n$ matrix A and $\mathbf{x} \in \mathbf{R}^n$, $A\mathbf{x}$ is linear combination of columns with x_j as scalar coefficient for jth column	Theorem 13.3.4
jth column of A is $A\mathbf{e}_j$	matrix-vector products $A\mathbf{e}_1, \dots, A\mathbf{e}_n$ are columns of A	Theorem 13.4.5
"symmetry" of $(D(\nabla f))(\mathbf{a})$	ij-entry and ji-entry of $(D(\nabla f))(\mathbf{a})$ coincide	Example 13.5.6

Skill	**Location in text**
express affine or linear function in terms of matrix-vector product	Exs. 13.3.5, 13.3.7, 13.3.9, 13.3.12
for linear $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$, know that associated $m \times n$ matrix A has jth column $\mathbf{f}(\mathbf{e}_j)$ (this is a **very** important fact!)	Theorem 13.4.5
compute derivative matrix symbolically and numerically	Examples 13.5.2–13.5.5
compute best linear approximation to $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ at given $\mathbf{a} \in \mathbf{R}^n$	Example 13.5.9

14. Linear transformations and matrix multiplication

In this chapter, we will introduce a fundamental new idea: how to *multiply* matrices. This has some unexpected features, such as $AB \neq BA$, but it turns out to cleanly encode many interesting phenomena.

By the end of this chapter, you should be able to:

- characterize a linear function $\mathbf{R}^n \to \mathbf{R}^m$ in terms of its interaction with vector addition and scalar multiplication;
- compute the matrix of a rotation in $\mathbf{R}^2$ and of a rotation around a coordinate axis in $\mathbf{R}^3$;
- compose two linear functions using matrix multiplication.

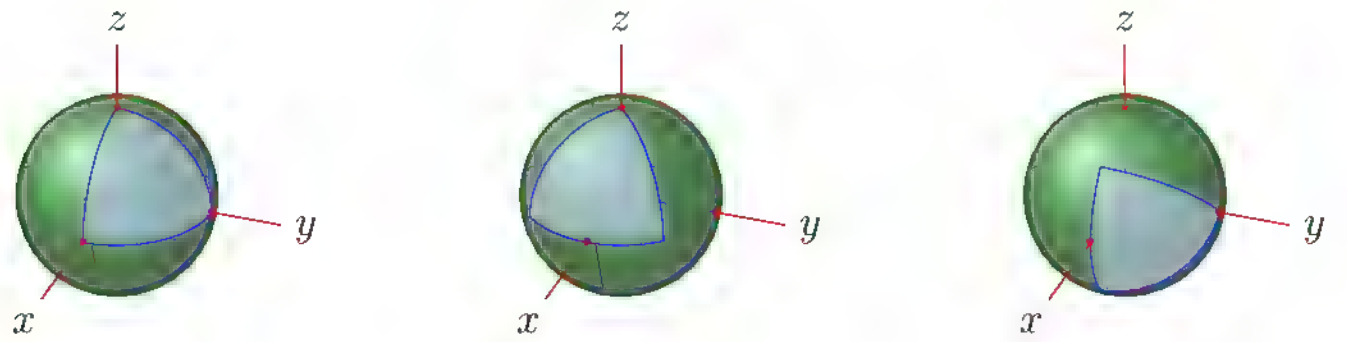
14.1. Rotations and linear functions. If you rotate a steering wheel by 30 degrees counterclockwise, and then by 10 degrees clockwise ("counterclockwise by -10 degrees"), you have rotated the wheel by $30 - 10 = 20$ degrees counterclockwise.

In general, the overall effect of applying a counterclockwise rotation by an angle θ_1 and then a counterclockwise rotation by an angle θ_2 is such a rotation: counterclockwise by the angle $\theta_1 + \theta_2$ (even if one or both of the θ_j's is "negative"). Note in particular that *the order in which we apply the* 2*-dimensional rotations doesn't matter* for the final outcome since the angle sums $\theta_1 + \theta_2$ and $\theta_2 + \theta_1$ are equal.

When we consider an analogue in 3 dimensions, the situation is more subtle (and so more interesting). Think of the following operations you can do to the surface of a ball:

- operation R_1: rotate the ball by -45 degrees around the vertical axis through its center;
- operation R_2: rotate the ball by 45 degrees around a specified horizontal axis through its center.

These operations are shown in Figure 14.1.1 (where the specified horizontal axis for R_2 is the y-axis); the colors in the figure are included to illustrate how the rotation affects the ball.



A ball.

Operation R_1:
The ball rotated by $-45°$ around z-axis.

Operation R_2:
The ball rotated by $45°$ around y-axis.

FIGURE 14.1.1. Two different rotations of a ball

What if you do operation R_1 first and then operation R_2? What if you do operation R_2 first and then operation R_1? As shown in Figure 14.1.2, the overall effect of these is not the same! (For the analogue with 90-degree rotations, you may want to try it with an actual ball by marking points on the ball with a pencil or some Hagoromo chalk.)

Operation R_1 followed by operation R_2. Operation R_2 followed by operation R_1.

FIGURE 14.1.2. The order of operations makes a difference for 3-dimensional rotations

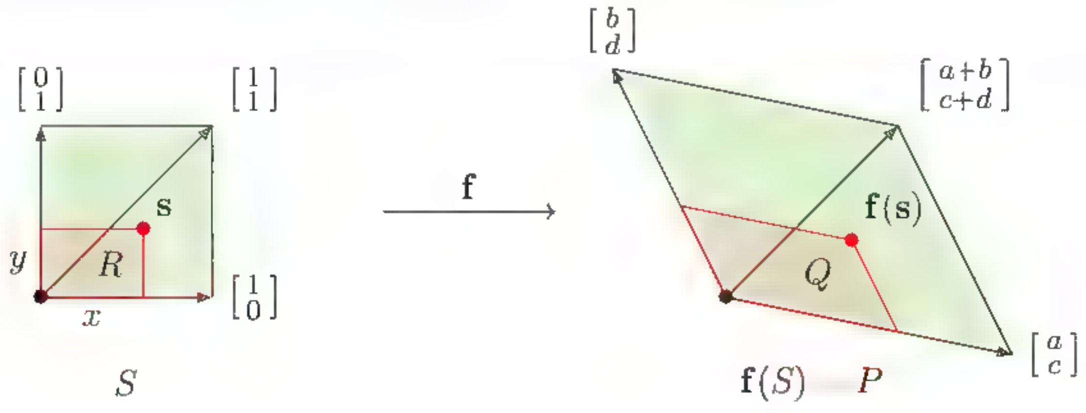
A good understanding of rotations is useful in many fields, such as robotics (how do you rotate a joint to move an arm to the desired position?), computer graphics (how do you rotate a scene?), and biology (how do you manipulate a computer model of the human brain?). The key to working with rotations is that they are *linear* in the sense defined in Section 13.2. This linearity has to be explained. We will do this in Section 14.4. To prepare for it, let's revisit the definition of linear and affine functions.

Definition 14.1.1. A function $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ is called :

- a *linear function*, or a *linear transformation*, if there is an $m \times n$ matrix A for which $\mathbf{f}(\mathbf{x}) = A\mathbf{x}$ for all vectors $\mathbf{x} \in \mathbf{R}^n$ (so the jth column of A is $A\mathbf{e}_j = \mathbf{f}(\mathbf{e}_j)$),
- an *affine function*, or an *affine transformation*, if there is an $m \times n$ matrix A and a vector $\mathbf{b} \in \mathbf{R}^m$ for which $\mathbf{f}(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$ for all vectors $\mathbf{x} \in \mathbf{R}^n$.

This is the same as our old definitions from Section 13.2, recast using the language of matrices; see Examples 13.2.2 and 13.3.12 for explicit examples. Before we give some more examples, including real-world applications of affine transformations, we want to discuss how to visualize the effect of the linear transformation $\mathbf{f} : \mathbf{R}^2 \to \mathbf{R}^2$ associated with a matrix $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ whose columns span different lines through $\mathbf{0}$. Let S be the "unit square" $\{(x, y) \in \mathbf{R}^2 : 0 \leq x, y \leq 1\}$ (on the left in Figure 14.1.3), so $\mathbf{R}^2$ is tiled by copies of S laid out in all directions (like a bathroom floor). Let $\mathbf{f}(S)$ denote the output of $\mathbf{f}$ on S (i.e., the collection of points $\mathbf{f}(\mathbf{s})$ for $\mathbf{s} \in S$), also called the *image* of S under $\mathbf{f}$. The image $\mathbf{f}(S)$ is a parallelogram (on the right in Figure 14.1.3). This helps to visualize the effect of $\mathbf{f}$ on $\mathbf{R}^2$ due to two facts:

Linearity Principle: for $c_1, c_2 \in \mathbf{R}$ and $\mathbf{v}_1, \mathbf{v}_2 \in \mathbf{R}^2$ we have $\mathbf{f}(c_1\mathbf{v}_1 + c_2\mathbf{v}_2) = c_1\mathbf{f}(\mathbf{v}_1) + c_2\mathbf{f}(\mathbf{v}_2)$.
Tiling Principle: $\mathbf{f}$ transforms the tiling of $\mathbf{R}^2$ by copies of S into a tiling of $\mathbf{R}^2$ by copies of $\mathbf{f}(S)$.



$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\mathbf{s}$
y
R
$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$
x
S
$\mathbf{f}$
$\begin{bmatrix} b \\ d \end{bmatrix}$
$\begin{bmatrix} a+b \\ c+d \end{bmatrix}$
$\mathbf{f}(\mathbf{s})$
Q
$\begin{bmatrix} a \\ c \end{bmatrix}$
$\mathbf{f}(S)$
P

FIGURE 14.1.3. The linear transformation $\mathbf{f}$ takes the unit square S onto a parallelogram P. The effect of $\mathbf{f}$ is shown on a typical point $\mathbf{s} = (x, y)$ inside S, and $\mathbf{f}(R) = Q$.

The Linearity Principle (which we'll see in a wider context in Section 14.2) is a 1-line calculation:

$$\mathbf{f}(c_1\mathbf{v}_1 + c_2\mathbf{v}_2) = A(c_1\mathbf{v}_1 + c_2\mathbf{v}_2) = A(c_1\mathbf{v}_1) + A(c_2\mathbf{v}_2) = c_1(A\mathbf{v}_1) + c_2(A\mathbf{v}_2) = c_1\mathbf{f}(\mathbf{v}_1) + c_2\mathbf{f}(\mathbf{v}_2).$$

This underlies why $\mathbf{f}(S)$ is the parallelogram with vertices as in Figure 14.1.3 (using the parallelogram law for vector addition), as well as why $\mathbf{f}(R) = Q$ in the setting of that Figure. Proofs of these facts, as well as a proof of the Tiling Principle, are in Section 14.6 for those who are interested. Let's use this to work out examples to visualize the linear transformations $\mathbf{R}^2 \to \mathbf{R}^2$ associated with some 2×2 matrices.

Example 14.1.2. Consider $f : \mathbf{R}^2 \to \mathbf{R}^2$ given by the matrix $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$. By Figure 14.1.3, $f(S)$ is the parallelogram with edges $f\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ (first column of A) and $f\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ (second column of A), as shown in Figure 14.1.4. By the Tiling Principle, f transforms the usual tiling of $\mathbf{R}^2$ by unit squares into the tiling by copies of $f(S)$ laid out in all directions parallel to the edges of $f(S)$.

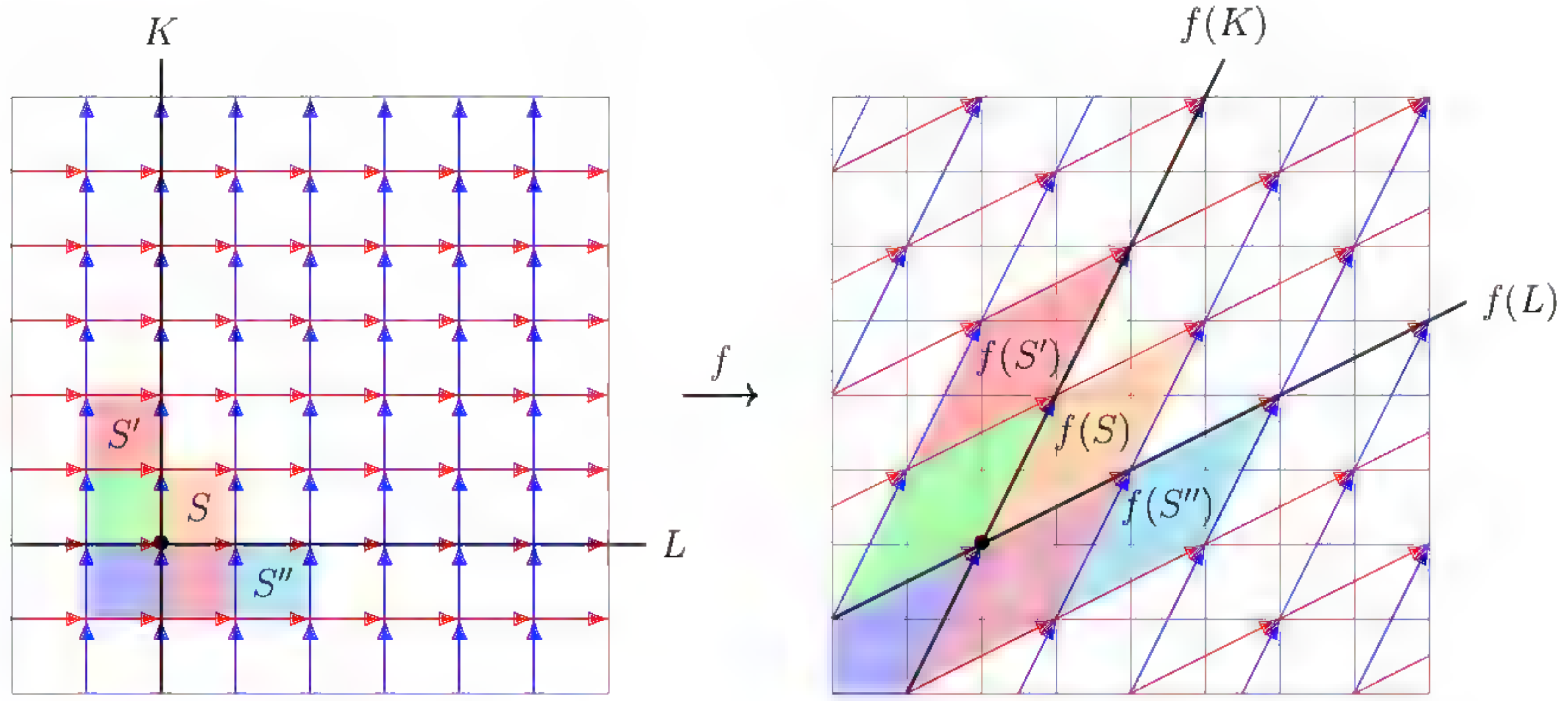


FIGURE 14.1.4. The image $f(S)$ of the unit square S is a parallelogram, and the collection of unit squares in the tiling on the left is carried by f to parallelograms that give a tiling on the right, as illustrated with nearby unit squares S' and S''.

In Figure 14.1.4, the coordinate axes on the left are the black lines K and L through edges of the orange unit square S which meet at the black dot that is the origin on the left, and f carries those onto the black lines $f(K)$ and $f(L)$ through edges of the orange parallelogram $f(S)$ on the right (meeting at the black dot corresponding to the origin, since f carries the origin to the origin).

The tiling on the right in Figure 14.1.4 expresses the *parametric form* of $\mathbf{R}^2$ using the vectors $\mathbf{e} = f\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ (first column of A) and $\mathbf{e}' = f\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ (second column of A). In particular, the Linearity Principle

$$t\mathbf{e} + t'\mathbf{e}' = tf\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) + t'f\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = f\left(t\begin{bmatrix} 1 \\ 0 \end{bmatrix} + t'\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = f\left(\begin{bmatrix} t \\ t' \end{bmatrix}\right)$$

says that the point in parametric form $t\mathbf{e} + t'\mathbf{e}'$ is the output of f on $\begin{bmatrix} t \\ t' \end{bmatrix}$. In the special case that t and t' are integers, which corresponds to the corners of the parallelograms in the tiling on the right side of

Figure 14.1.4 (due to the geometric interpretation of vector addition and subtraction), this is telling us algebraically the fact we can see visually that those corners are the output of f on the corners of the unit square tiling on the left (as these latter corners are the points of $\mathbf{R}^2$ with integer coordinates). ■

Example 14.1.3. Let $T_c : \mathbf{R}^2 \to \mathbf{R}^2$ be $T_c(\mathbf{x}) = M\mathbf{x}$ for $M = \begin{bmatrix} 1 & c \\ 0 & 1 \end{bmatrix}$. The output $T_c(S)$ of T_c on S is the parallelogram with edges $T_c(\mathbf{e}_1) = M \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ (first column of M) and $T_c(\mathbf{e}_2) = M \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} c \\ 1 \end{bmatrix}$ (second column of M). This has the same base $\mathbf{e}_1$ as S and the same height (namely: 1) as S, but it has a horizontal displacement by c units along the top; see Figure 14.1.5.

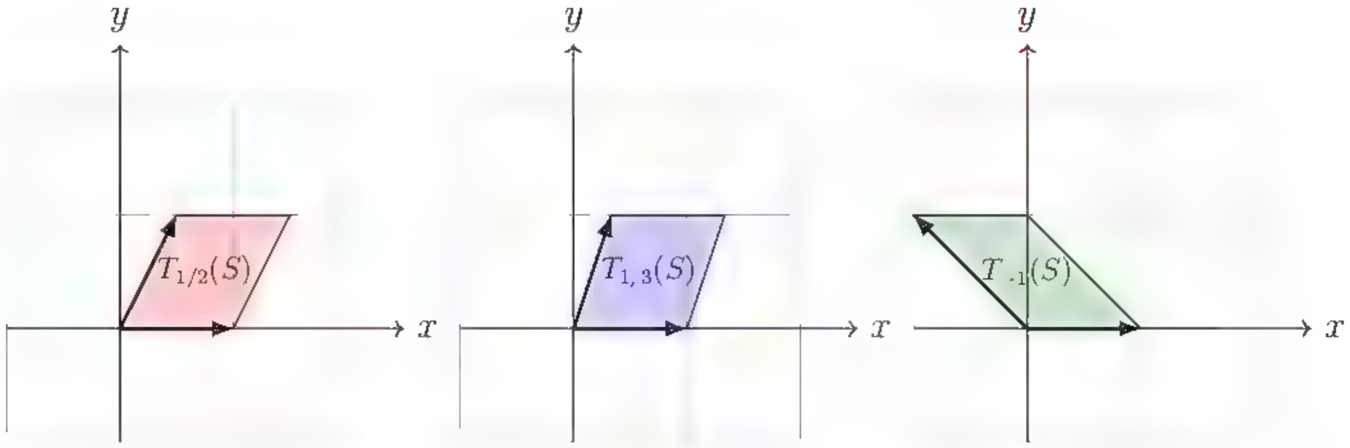


FIGURE 14.1.5. The image of the unit square S under T_c for $c = 1/2, 1/3, -1$.

The transformations T_c are called *shearing transformations*: they distort shape, but they *do not affect area* (think of pushing the side of a deck of cards, with more pushing near the top than the bottom). By the Tiling Principle, T_c transforms the usual tiling of $\mathbf{R}^2$ by unit squares into the tiling by copies of $T_c(S)$ laid out in all directions parallel to the edges of $T_c(S)$. This amounts to "pushing" the unit square tiling in the horizontal direction to tilt it from being vertically upright without affecting area. ■

Example 14.1.4. When MRI is used to create a 3-dimensional image of someone's brain and doctors use the image for medical diagnosis, the computers use many affine transformations $f(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$. The matrix A accounts for rotation of the image in space, as well as dilation for "zooming in/out" on the image, and the vector $\mathbf{b}$ is a displacement vector that accounts for spatial translation of the image. ■

Example 14.1.5. Consider the problem of an autonomous car correctly identifying and reading traffic signs (such as stop signs, yield signs, speed limit signs, etc.) Typically the camera on the car will see the image of such a sign in a manner that is distorted due to the viewing angle and environmental conditions (such as how far the sign is from the road, etc.): a circular sign may appear as an ellipse, an equilateral triangle sign may appear as a scalene triangle, and a rectangular sign may appear as a parallelepiped. Specific writing (such as a speed limit, or an arrow pointing in some direction) is then similarly distorted.

By making visual measurements of the main geometric shapes in the image, a computer can produce a 2×2 matrix A that encodes shearing and rotations and a vector $\mathbf{b} \in \mathbf{R}^2$ that encodes recentering so that $A\mathbf{x} + \mathbf{b}$ undoes the distortion to compare the sign to a database of "standard" possibilities. This process (which may also use color information) can then clarify specialized writing in the sign too. ■

Example 14.1.6. The same math problem as in Example 14.1.4 arises in image alignment: merging multiple images of the same object (a building, skyline of a city, etc.) from different perspectives into a single overall image. By identifying specific points $\mathbf{c}_1, \dots, \mathbf{c}_m \in \mathbf{R}^2$ in the first image with their counterparts $\mathbf{b}_1, \dots, \mathbf{b}_m \in \mathbf{R}^2$ in the second image, to merge the images we seek a 2×2 distortion matrix $\mathbf{X}$ and a displacement vector $\mathbf{y}$ so that $\mathbf{b}_j = \mathbf{X}\mathbf{c}_j + \mathbf{y}$ for all j. The affine transformation $f(\mathbf{v}) = \mathbf{X}\mathbf{v} + \mathbf{y}$ can then be used to determine the "overlap" of the images and then merge them.

Due to rounding errors, there are typically no $\mathbf{X}$ and $\mathbf{y}$ that make the m conditions $\mathbf{b}_j = \mathbf{X}\mathbf{c}_j + \mathbf{y}$ exactly hold for all j (and for large m, this system of conditions on the 4 entries of $\mathbf{X}$ and the 2 entries of $\mathbf{y}$ is vastly "overdetermined": far more than 6 conditions on these 6 parameters). So we seek $\mathbf{X}$ and $\mathbf{y}$ that minimize the total error $\sum_j \|\mathbf{b}_j - (\mathbf{X}\mathbf{c}_j + \mathbf{y})\|^2$ in a "least squares" sense that generalizes Chapter 7. The technique to solve many such problems will be given in Theorem 22.5.4. ■

Affine transformations are the *multivariable analogues* of the functions $mx + b$ that you know from tangent-line approximations in single-variable calculus. A fundamental idea of multivariable calculus is:

any function $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^p$ is well-approximated near any $\mathbf{a} \in \mathbf{R}^n$ by an affine function.

For $n = p = 1$ this says any $f : \mathbf{R} \to \mathbf{R}$ can be approximated near any $a \in \mathbf{R}$ by some $mx + b$ (an affine function of 1 variable); it is the tangent-line approximation with the slope m taken to be $f'(a)$.

14.2. Linear functions are those which respect addition and scalar multiplication. We have defined the concept of "linear function" by a type of formula. It is also useful to characterize them by some properties. (Why? See Section 14.4.) This is accomplished by the following result, which says that *linear functions* $\mathbf{R}^n \to \mathbf{R}^m$ *are precisely the ones which respect scalar multiplication and vector addition.*

Theorem 14.2.1. A function $\mathbf{g} : \mathbf{R}^n \to \mathbf{R}^m$ is linear precisely when it respects the vector operations:

$$\mathbf{g}(c\mathbf{x}) = c\mathbf{g}(\mathbf{x}), \quad \mathbf{g}(\mathbf{x}+\mathbf{y}) = \mathbf{g}(\mathbf{x}) + \mathbf{g}(\mathbf{y}) \tag{14.2.1}$$

for all scalars $c \in \mathbf{R}$ and vectors $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$.

If $\mathbf{g} : \mathbf{R}^n \to \mathbf{R}^m$ and $\mathbf{h} : \mathbf{R}^p \to \mathbf{R}^n$ are linear, then so is the composition $\mathbf{g} \circ \mathbf{h} : \mathbf{R}^p \to \mathbf{R}^m$.

The proof of Theorem 14.2.1 is given in Section 14.6, for those who are interested. Property (14.2.1) is the crucial identifying feature of linear functions. *Affine functions* $\mathbf{h}(\mathbf{x}) = A\mathbf{x}+\mathbf{b}$ *with* $\mathbf{b} \neq \mathbf{0}$ *do not satisfy this property.* For instance: $\mathbf{h}(5\mathbf{x}) = A(5\mathbf{x}) + \mathbf{b} = 5(A\mathbf{x}) + \mathbf{b}$ and $5\,\mathbf{h}(\mathbf{x}) = 5(A\mathbf{x} + \mathbf{b}) = 5(A\mathbf{x}) + 5\mathbf{b}$, so $\mathbf{h}(5\mathbf{x}) \neq 5\,\mathbf{h}(\mathbf{x})$ because $\mathbf{b} \neq 5\mathbf{b}$ when $\mathbf{b} \neq \mathbf{0}$ (compare lengths of $\mathbf{b}$ and $5\mathbf{b}$ for nonzero $\mathbf{b}$). Likewise,

$$\mathbf{h}(\mathbf{x}+\mathbf{y}) = A(\mathbf{x}+\mathbf{y}) + \mathbf{b} = A\mathbf{x} + A\mathbf{y} + \mathbf{b}, \;\; \mathbf{h}(\mathbf{x}) + \mathbf{h}(\mathbf{y}) = (A\mathbf{x}+\mathbf{b}) + (A\mathbf{y}+\mathbf{b}) = A\mathbf{x} + A\mathbf{y} + 2\mathbf{b},$$

so $\mathbf{h}(\mathbf{x}+\mathbf{y}) \neq \mathbf{h}(\mathbf{x}) + \mathbf{h}(\mathbf{y})$ when $\mathbf{b}$ is nonzero (since $\mathbf{b} \neq 2\mathbf{b}$ for such $\mathbf{b}$).

The failure of (14.2.1) for general affine functions is one of the reasons why linear functions are more fundamental than affine functions. Here is an interesting immediate application of Theorem 14.2.1:

Example 14.2.2. For any nonzero linear subspace V of $\mathbf{R}^n$ we claim that $\mathbf{Proj}_V : \mathbf{R}^n \to \mathbf{R}^n$ is linear (so it is given by some $n \times n$ matrix!). This may be surprising since the geometric definition of $\mathbf{Proj}_V(\mathbf{x})$ in terms of distance minimization doesn't help for understanding how $\mathbf{Proj}_V$ interacts with vector addition and scalar multiplication. There is another way to describe $\mathbf{Proj}_V$ that makes more contact with vector algebra: if $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ is an orthogonal basis of V then formula (6.2.1) gives the explicit formula

$$\mathbf{Proj}_V(\mathbf{x}) = \sum_{i=1}^{k} \left(\frac{\mathbf{x} \cdot \mathbf{w}_i}{\mathbf{w}_i \cdot \mathbf{w}_i} \right) \mathbf{w}_i.$$

So the linearity of $\mathbf{Proj}_V$ can be rephrased as: does this summation expression behave well with respect to addition and scalar multiplication in $\mathbf{x}$?

Yes: for all $\mathbf{x},\mathbf{y} \in \mathbf{R}^n$ we have $(\mathbf{x}+\mathbf{y}) \cdot \mathbf{w} = \mathbf{x} \cdot \mathbf{w} + \mathbf{y} \cdot \mathbf{w}$ for every $\mathbf{w} \in \mathbf{R}^n$, so

$$\begin{aligned}\mathbf{Proj}_V(\mathbf{x}+\mathbf{y}) = \sum_{i-1}^{k} \left(\frac{(\mathbf{x}+\mathbf{y}) \cdot \mathbf{w}_i}{\mathbf{w}_i \cdot \mathbf{w}_i} \right) \mathbf{w}_i &= \sum_{i=1}^{k} \left(\frac{\mathbf{x} \cdot \mathbf{w}_i + \mathbf{y} \cdot \mathbf{w}_i}{\mathbf{w}_i \cdot \mathbf{w}_i} \right) \mathbf{w}_i \\ &= \sum_{i=1}^{k} \left(\frac{\mathbf{x} \cdot \mathbf{w}_i}{\mathbf{w}_i \cdot \mathbf{w}_i} \right) \mathbf{w}_i + \sum_{i=1}^{k} \left(\frac{\mathbf{y} \cdot \mathbf{w}_i}{\mathbf{w}_i \cdot \mathbf{w}_i} \right) \mathbf{w}_i \\ &= \mathbf{Proj}_V(\mathbf{x}) + \mathbf{Proj}_V(\mathbf{y}).\end{aligned}$$

Likewise, for any $\mathbf{x} \in \mathbf{R}^n$ and $c \in \mathbf{R}$ we have $(c\mathbf{x}) \cdot \mathbf{w} = c(\mathbf{x} \cdot \mathbf{w})$ for every $\mathbf{w} \in \mathbf{R}^n$, so

$$\mathbf{Proj}_V(c\mathbf{x}) = \sum_{i=1}^{k} \left(\frac{(c\mathbf{x}) \cdot \mathbf{w}_i}{\mathbf{w}_i \cdot \mathbf{w}_i} \right) \mathbf{w}_i = \sum_{i=1}^{k} \left(\frac{c(\mathbf{x} \cdot \mathbf{w}_i)}{\mathbf{w}_i \cdot \mathbf{w}_i} \right) \mathbf{w}_i = c \sum_{i=1}^{k} \left(\frac{\mathbf{x} \cdot \mathbf{w}_i}{\mathbf{w}_i \cdot \mathbf{w}_i} \right) \mathbf{w}_i = c\,\mathbf{Proj}_V(\mathbf{x}).$$

This concludes the verification that $\mathbf{Proj}_V$ is linear. Actually, not quite: we swept something under the rug. Can you see the gap in our argument? The gap is that we took it for granted that V *has* an orthogonal basis (so formula (6.2.1) is actually applicable to V!). Up to now we don't know that linear subspaces V always have orthogonal bases except when $\dim V = 1, 2$. This hole in our knowledge will be filled in Chapter 19, so at that time the above argument will be complete; let's not worry about it for now.

What is the $n \times n$ matrix for $\mathbf{Proj}_V$? As for any linear function $\mathbf{R}^n \to \mathbf{R}^n$, its jth column is its effect on $\mathbf{e}_j$, which is to say its jth column is

$$\mathbf{Proj}_V(\mathbf{e}_j) = \sum_{i=1}^{k} \left(\frac{\mathbf{e}_j \cdot \mathbf{w}_i}{\mathbf{w}_i \cdot \mathbf{w}_i} \right) \mathbf{w}_i \tag{14.2.2}$$

upon choosing an orthogonal basis $\{\mathbf{w}_1, \dots, \mathbf{w}_k\}$ of V (the choice of basis is a tool to compute $\mathbf{Proj}_V(\mathbf{e}_j)$; the output of the summation formula is *independent* of the choice!). To give a numerical example, consider V defined in Example 5.2.3 as the span of specific $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \in \mathbf{R}^5$ in (5.2.1) whose entries are tiny integers. In Example 5.2.3 we gave an explicit orthogonal triple of nonzero vectors in V,

$$\mathbf{w}_1 = \begin{bmatrix} 1 \\ 0 \\ 3 \\ 2 \\ 1 \end{bmatrix}, \quad \mathbf{w}_2 = \begin{bmatrix} 1 \\ 3 \\ 0 \\ -4 \\ 7 \end{bmatrix}, \quad \mathbf{w}_3 = \begin{bmatrix} -33 \\ 201 \\ -75 \\ 132 \\ -6 \end{bmatrix}, \tag{14.2.3}$$

that we verified is an orthogonal basis of V (the $\mathbf{w}_i$'s are found using the Gram–Schmidt process to be discussed in Chapter 19). Hence, the 5×5 matrix M of $\mathbf{Proj}_V : \mathbf{R}^5 \to \mathbf{R}^5$ has jth column given by (14.2.2) with $k = 3$ using (14.2.3).

Since $\mathbf{w}_1 \cdot \mathbf{w}_1 = 15$, $\mathbf{w}_2 \cdot \mathbf{w}_2 = 75$, $\mathbf{w}_3 \cdot \mathbf{w}_3 = 64575 = 15 \times 4305 = 45 \times 1435$, and $\mathbf{e}_j \cdot \mathbf{w}$ is the jth entry of $\mathbf{w}$ for any $\mathbf{w} \in \mathbf{R}^5$, the columns of M can be computed quickly on a computer, yielding

$$M = \frac{1}{1435} \begin{bmatrix} 139 & -90 & 342 & 18 & 234 \\ -90 & 1070 & -335 & 360 & 375 \\ 342 & -335 & 986 & 354 & 297 \\ 18 & 360 & 354 & 1076 & -362 \\ 234 & 325 & 297 & -362 & 1034 \end{bmatrix}. \tag{14.2.4}$$

Thus, even for a subspace V defined as the span of vectors (such as $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ in (5.2.1)) whose entries are tiny integers, the matrix for $\mathbf{Proj}_V$ can be rather "complicated"! ■

In Section 14.4 we will use Theorem 14.2.1 to see that rotations around the origin in $\mathbf{R}^2$ and $\mathbf{R}^3$ are linear, so then it is meaningful to compute a matrix encoding the effect of such operations. This will enable us to algebraically interpret the rotation behavior in Section 14.1.

Example 14.2.3. A "linear" circuit is one whose output (e.g., currents in various parts, which we want to know) is a linear function of its input (e.g., voltage differences in various parts, which *we control*). Concretely, these are the circuits involving only resistors, capacitors, and inductors; they account for all of the circuits studied in introductory physics (whereas circuits involving components such as transistors or diodes are *not* linear). The analysis of *every* linear circuit, no matter how big or complicated, *always* can be done systematically via (possibly very high-dimensional!) linear algebra: see item (2) in Section 21.1 for further discussion and precise references. ■

14.3. Composing linear transformations and matrix multiplication. Composing linear transformations brings us to one of the key operations of the course: *matrix multiplication*. The definition below may initially look like a mouthful, but after we work out some examples we'll see that it isn't too bad.

Definition 14.3.1. Let A be an $m \times n$ matrix and B an $n \times p$ matrix as follows:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1p} \\ b_{21} & b_{22} & \dots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{np} \end{bmatrix}$$

Let $T_A : \mathbf{R}^n \to \mathbf{R}^m$ and $T_B : \mathbf{R}^p \to \mathbf{R}^n$ be the linear transformations with respective matrices A and B (i.e., $T_A(\mathbf{x}) = A\mathbf{x}$ for $\mathbf{x} \in \mathbf{R}^n$, and $T_B(\mathbf{y}) = B\mathbf{y}$ for $\mathbf{y} \in \mathbf{R}^p$), so the composition $T_A \circ T_B : \mathbf{R}^p \to \mathbf{R}^m$ is a linear transformation (by Theorem 14.2.1).

The $m \times p$ matrix of $T_A \circ T_B$ is called the *matrix product* of A and B, and is denoted AB.

Theorem 14.3.2. The entries of AB are the dot products of *rows* of A with *columns* of B: if we write

$$A = \begin{bmatrix} — & \mathbf{a}_1 & — \\ — & \mathbf{a}_2 & — \\ & \vdots & \\ — & \mathbf{a}_m & — \end{bmatrix}, \quad B = \begin{bmatrix} | & | & & | \\ \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_p \\ | & | & & | \end{bmatrix}$$

with rows $\mathbf{a}_i \in \mathbf{R}^n$ and columns $\mathbf{b}_j \in \mathbf{R}^n$, then we have

$$AB = \begin{bmatrix} \mathbf{a}_1 \cdot \mathbf{b}_1 & \mathbf{a}_1 \cdot \mathbf{b}_2 & \dots & \mathbf{a}_1 \cdot \mathbf{b}_p \\ \mathbf{a}_2 \cdot \mathbf{b}_1 & \mathbf{a}_2 \cdot \mathbf{b}_2 & \dots & \mathbf{a}_2 \cdot \mathbf{b}_p \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_m \cdot \mathbf{b}_1 & \mathbf{a}_m \cdot \mathbf{b}_2 & \dots & \mathbf{a}_m \cdot \mathbf{b}_p \end{bmatrix} = \begin{bmatrix} | & | & & | \\ A\mathbf{b}_1 & A\mathbf{b}_2 & \cdots & A\mathbf{b}_p \\ | & | & & | \end{bmatrix}.$$

Written out more explicitly,

$$ij\text{-entry of } AB = \sum_{k=1}^{n} a_{ik} b_{kj}. \tag{14.3.1}$$

A proof of (14.3.1) based on considerations with linearity is given in Section 14.6 for those who are interested (ignore it if you prefer). In the next chapter we will investigate the properties of matrix multiplication in detail, explaining in particular why this deserves the name "multiplication". **It only makes sense to form AB when the number of columns of A is the same as the number of rows of B**

(so the dot product $\mathbf{a}_i \cdot \mathbf{b}_j$ of a row $\mathbf{a}_i$ of A and a column $\mathbf{b}_j$ of B makes sense); this requirement expresses the fact that it only makes sense to form $T_A \circ T_B$ when the output of T_B is an input for T_A.

To get a feeling for matrix multiplication, let's work out some examples. We begin with some that get us acclimated to the process, and in Sections 14.4 and 14.5 we discuss geometric examples. (As another geometric example, in the setting of Example 14.2.2 necessarily $\mathbf{Proj}_V \circ \mathbf{Proj}_V = \mathbf{Proj}_V$ [why?], so M in (14.2.4) satisfies $M^2 = M$!) In each case below, *please check the calculations for yourself.*

Example 14.3.3. For the matrices $A = \begin{bmatrix} 2 & 5 \\ 1 & 1 \end{bmatrix}$ and $B = \begin{bmatrix} 3 & 1 \\ 0 & -1 \end{bmatrix}$, computing dot products of rows of A against columns of B yields that

$$AB = \begin{bmatrix} \mathbf{2} & \mathbf{5} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 3 & \mathbf{1} \\ 0 & \mathbf{1} \end{bmatrix} = \begin{bmatrix} 6 & \mathbf{-3} \\ 3 & 0 \end{bmatrix} = \begin{bmatrix} A\begin{bmatrix} 3 \\ 0 \end{bmatrix} & A\begin{bmatrix} 1 \\ -1 \end{bmatrix} \end{bmatrix}$$

(e.g., the **first** row of A and **second** column of B have dot product -3 that is the **(1,2)**-entry in the output).

Here is an alternative way to write out the preceding calculation that takes up more space on the page but is much more efficient in practice for visually organizing the work (and is nicely illustrated on the Wikipedia page for matrix multiplication). To compute AB, we write out the task in the following way:

$$\begin{array}{cc} & \begin{bmatrix} 3 & \mathbf{1} \\ 0 & \mathbf{1} \end{bmatrix} \\ \begin{bmatrix} \mathbf{2} & \mathbf{5} \\ 1 & 1 \end{bmatrix} & \begin{bmatrix} * & \boldsymbol{*} \\ * & * \end{bmatrix} \end{array}$$

with A on the left (just as it appears in "AB"), the other matrix B moved up, and the $*$'s as entries to be determined to give AB. (We are writing $*$'s in this book for expository purposes; when you do this yourself you should leave empty spaces in which you'll soon fill in numbers.) The rule is this: to fill in a specific entry $*$, form the dot product of the row (in the matrix to the left) on the same horizontal line as the chosen $*$ and the column (in the matrix on top) on the same vertical line as the chosen $*$.

For example, the boldface asterisk is the dot product of the row and column (both in boldface) displayed in a visually appealing way relative to that asterisk (and likewise for the others)! We fill in the entries essentially by inspection

$$\begin{array}{cc} & \begin{bmatrix} 3 & \mathbf{1} \\ 0 & \mathbf{-1} \end{bmatrix} \\ \begin{bmatrix} \mathbf{2} & \mathbf{5} \\ 1 & 1 \end{bmatrix} & \begin{bmatrix} 6+0 & \mathbf{2-5} \\ 3+0 & 1-1 \end{bmatrix} \end{array},$$

so the product is $\begin{bmatrix} 6 & -3 \\ 3 & 0 \end{bmatrix}$. See (14.3.2) for the analogous diagram to compute MN for a 2×4 matrix M and a 4×3 matrix N.

Switching to dot products of rows of B against columns of A, the product BA in the other order is obtained by filling in the array of $*$'s in accordance with the diagram

$$\begin{array}{cc} & \begin{bmatrix} 2 & 5 \\ 1 & 1 \end{bmatrix} \\ \begin{bmatrix} 3 & 1 \\ 0 & -1 \end{bmatrix} & \begin{bmatrix} * & * \\ * & * \end{bmatrix} \end{array}$$

Staring at this, we read off the number for each $*$:

$$BA = \begin{bmatrix} 6+1 & 15+1 \\ 0-1 & 0-1 \end{bmatrix} = \begin{bmatrix} 7 & 16 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} B\begin{bmatrix} 2 \\ 1 \end{bmatrix} & B\begin{bmatrix} 5 \\ 1 \end{bmatrix} \end{bmatrix}.$$

So in particular, $AB \neq BA$! ■

Example 14.3.4. Let $A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ (so T_A crushes the y-axis into $\mathbf{0}$ since $T_A(\mathbf{e}_2) = A\mathbf{e}_2$ is the second column $\mathbf{0}$ of A, and T_A has no effect on the x-axis since $T_A(\mathbf{e}_1) = A\mathbf{e}_1$ is the first column $\mathbf{e}_1$ of A), and $B = \begin{bmatrix} 0 & 0 \\ 2 & -3 \end{bmatrix}$ (so T_B crushes $\mathbf{R}^2$ into the y-axis since the columns $T_B(\mathbf{e}_1) = B\mathbf{e}_1$ and $T_B(\mathbf{e}_2) = B\mathbf{e}_2$ of B are both contained in the y-axis). To compute AB we form the diagram

$$\begin{array}{cc} & \begin{bmatrix} 0 & 0 \\ 2 & -3 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} * & * \\ * & * \end{bmatrix} \end{array}$$

We see that all $*$'s are equal to $0 + 0 = 0$, which is to say $AB = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$. Such vanishing can also be seen from the geometric descriptions of T_A and T_B since $T_{AB} = T_A \circ T_B$ and applying T_B sends everything into the y-axis yet that line crushed to $\{\mathbf{0}\}$ by T_A.

On the other hand, to compute BA we form the diagram

$$\begin{array}{cc} & \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \\ 2 & -3 \end{bmatrix} & \begin{bmatrix} * & * \\ * & * \end{bmatrix} \end{array}$$

and thereby read off that $BA = \begin{bmatrix} 0+0 & 0+0 \\ 2+0 & 0+0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix}$, which is nonzero!

Since $AB = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = A \begin{bmatrix} 0 & 0 \\ 4 & -5 \end{bmatrix}$ with nonzero A yet $B \neq \begin{bmatrix} 0 & 0 \\ 4 & -5 \end{bmatrix}$ (and similarly with $(4, -5)$ replaced by *any* ordered pair (u_1, u_2) different from the bottom row $(2, -3)$ of B), we see that there is generally no "cancellation law" for matrix multiplication! ■

Example 14.3.5. The matrix $A = \begin{bmatrix} 3 & 0 \\ 0 & -2 \end{bmatrix}$ corresponds to stretching by 3 in the x-direction and stretching by 2 along with a reflection in the y-direction since the respective columns $A\mathbf{e}_1$ and $A\mathbf{e}_2$ of A are $3\mathbf{e}_1$ and $-2\mathbf{e}_2$. The matrix $B = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ corresponds to a 90-degree counterclockwise rotation (this will be explained in Section 14.4). To compute AB, we write out the diagram

$$\begin{array}{cc} & \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \\ \begin{bmatrix} 3 & 0 \\ 0 & -2 \end{bmatrix} & \begin{bmatrix} * & * \\ * & * \end{bmatrix} \end{array}$$

and thereby read off $AB = \begin{bmatrix} 0+0 & -3+0 \\ 0-2 & 0+0 \end{bmatrix} = \begin{bmatrix} 0 & -3 \\ -2 & 0 \end{bmatrix}$.

Since the effect of B swaps the coordinate axes yet the effects of A along the coordinate axes are quite different from each other, it is reasonable to expect that applying T_B before or after T_A yields different outcomes. And sure enough, we compute BA using the diagram

$$\begin{array}{cc} & \begin{bmatrix}3 & 0\\ 0 & -2\end{bmatrix} \\ \begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix} & \begin{bmatrix}* & *\\ * & *\end{bmatrix}\end{array}$$

which yields $BA = \begin{bmatrix}0+0 & 0+2\\ 3+0 & 0+0\end{bmatrix} = \begin{bmatrix}0 & 2\\ 3 & 0\end{bmatrix}$, so $BA \neq AB$. ■

You might find it disturbing that often $AB \neq BA$ (and that nonzero matrices can have product equal to a zero matrix, as in Example 14.3.4); we will see *many* more such examples below. However, this should only be bothersome after you have learned *why* we call it "multiplication" (and so denote it as we do); see (MM2) in Section 15.3. Since matrix multiplication is designed to express composition of linear functions, its "non-commutativity" is just an instance of the concrete fact that composing two functions in both possible orders (i.e., $f(g(x))$ and $g(f(x))$) often yields different outputs. For instance, in general:

(i) $2x+5 \neq 2(x+5)$ (composing doubling and adding 5),
(ii) $1/(x+7) \neq (1/x)+7$ (composing reciprocation and adding 7),
(iii) $(2x)^3 \neq 2x^3$ (composing cubing and doubling),
(iv) $e^{x/2} \neq e^x/2$ (composing exponentiation and halving).

Remark 14.3.6. The non-commutativity of matrix multiplication pervades *many* phenomena: algorithms for ranking webpages (see Remark D.2.1), computer vision, market risk analysis, and even the foundations of quantum mechanics. It may initially sound surprising that such a concept would show up anywhere, and it came as a huge surprise to physicists in the early 20th century to find that it is **absolutely central** in quantum mechanics (upon which all modern electronics rests). Max Born won the 1954 Nobel Prize for his work on the foundations of quantum mechanics, and in particular for a formulation in which matrix multiplication plays a key role. In his Nobel Prize lecture he said this:

> *"I could not take my mind off Heisenberg's multiplication rule, and after a week of intensive thought and trial I suddenly remembered an algebraic theory which I had learned from my teacher, Professor Rosanes, in Breslau. Such square arrays are well known to mathematicians and, in conjunction with a specific rule for multiplication, are called matrices. . . . I was as excited by this result as a sailor would be who, after a long voyage, sees from afar, the longed-for land."*[15]

There is a special situation where matrix multiplication is commutative, and it is more widely important than you might initially expect (as we'll see in the optional Chapter 27):

Proposition 14.3.7. For $n \times n$ *diagonal* matrices A and B, the product matrix AB is also diagonal and is obtained by multiplying the corresponding entries in A and B:

$$\begin{bmatrix}a_1 & 0 & \cdots & 0\\ 0 & a_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & a_n\end{bmatrix}\begin{bmatrix}b_1 & 0 & \cdots & 0\\ 0 & b_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & b_n\end{bmatrix} = \begin{bmatrix}a_1b_1 & 0 & \cdots & 0\\ 0 & a_2b_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & a_nb_n\end{bmatrix}.$$

In particular, $AB = BA$ for diagonal A and B since $a_ib_i = b_ia_i$ for every i.

---

[15] Schrödinger was rather less excited by the link to matrices, writing in [Sch5] that he "was discouraged, if not repelled, by what appeared to me as very difficult methods of transcendental algebra, and by the want of perspicuity."

Example 14.3.8. For the 3×3 "upper triangular" matrices

$$A = \begin{bmatrix} 1 & 3 & -5 \\ 0 & -2 & 1 \\ 0 & 0 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 1 & 1 \\ 0 & -1 & 3 \\ 0 & 0 & 2 \end{bmatrix},$$

to compute AB (using dot products of rows of A against columns of B) we analyze the diagram:

$$\begin{array}{cc} & \begin{bmatrix} 2 & 1 & 1 \\ 0 & -1 & 3 \\ 0 & 0 & 2 \end{bmatrix} \\ \begin{bmatrix} 1 & 3 & -5 \\ 0 & -2 & 1 \\ 0 & 0 & 4 \end{bmatrix} & \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \end{array}$$

This yields that $AB = \begin{bmatrix} 2 & 1-3 & 1+9-10 \\ 0 & 2 & -6+2 \\ 0 & 0 & 8 \end{bmatrix} = \begin{bmatrix} 2 & -2 & 0 \\ 0 & 2 & -4 \\ 0 & 0 & 8 \end{bmatrix}$ (again upper triangular). Similarly one finds (please check for yourself) $BA = \begin{bmatrix} 2 & 4 & -5 \\ 0 & 2 & 11 \\ 0 & 0 & 8 \end{bmatrix} \neq AB.$ $\blacksquare$

Example 14.3.9. For the 2×3 matrix A and 3×2 matrix B defined by $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ and $B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$, to compute AB involves filling in the diagram

$$\begin{array}{cc} & \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \\ \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} & \begin{bmatrix} * & * \\ * & * \end{bmatrix} \end{array}$$

Observe how the visual nature of the method automatically sets up the sought-after product matrix to have the correct size (2×2 for AB, in accordance with Definition 14.3.1). Filling in the $*$'s as the corresponding dot products, one finds (please check!) $AB = \begin{bmatrix} 1+6+15 & 2+8+18 \\ 4+15+30 & 8+20+36 \end{bmatrix} = \begin{bmatrix} 22 & 28 \\ 49 & 64 \end{bmatrix}$. Likewise, to compute BA involves filling in the diagram

$$\begin{array}{cc} & \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \\ \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} & \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix} \end{array}$$

where again the method automatically sets up the sought-after product matrix to have the correct size (3×3 for BA, in accordance with Definition 14.3.1). Filling in the $*$'s as the corresponding dot products, one finds (please check!)

$$BA = \begin{bmatrix} 1+8 & 2+10 & 3+12 \\ 3+16 & 6+20 & 9+24 \\ 5+24 & 10+30 & 15+36 \end{bmatrix} = \begin{bmatrix} 9 & 12 & 15 \\ 19 & 26 & 33 \\ 29 & 40 & 51 \end{bmatrix}.$$

$\blacksquare$

We won't write out the matrix multiplication diagrams anymore in this book, to save space; please feel free to keep using them for your own manual work with matrices of small size. It works even when the product matrix isn't square, such as AB with a 2×4 matrix A and a 4×3 matrix B: this appears as a 2×3 matrix (as it should):

$$\begin{array}{cc} & \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \\ b_{41} & b_{42} & b_{43} \end{bmatrix} \\ \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{bmatrix} & \begin{bmatrix} * & * & * \\ * & * & * \end{bmatrix} \end{array} \tag{14.3.2}$$

with entries that are dot products of 4-vectors. If one tries to write the corresponding diagram for the meaningless BA

$$\begin{array}{cc} & \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{bmatrix} \\ \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \\ b_{41} & b_{42} & b_{43} \end{bmatrix} & \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \end{array}$$

then the required dot products don't make sense (dot product of a 3-vector and a 2-vector), indicating that something has gone wrong.

Remark 14.3.10. Although the formula for matrix multiplication initially looks complicated, it is a very natural concept because it computes the *composition* of linear transformations (a fundamental operation on such transformations). You may however wonder about a more naive notion: for two $m \times n$ matrices A and B, we form the $m \times n$ matrix $A * B$ whose ij-entry is $a_{ij}b_{ij}$. This concept is called the *Hadamard product* and satisfies $A * B = B * A$; it only rarely arises in applications of linear algebra.

Remark 14.3.11. In applications to machine learning and many other mathematical models, multiplication of large matrices is an essential ingredient. In Section G.5 we discuss how knowledge of *properties* of matrix multiplication (such as associativity) provides crucial time savings in practical algorithms at the foundation of artificial intelligence. There are also some very clever algorithms for speeding up the computation of products of large matrices in general (reducing the task to computing products of "smaller" matrices), and special-purpose algorithms for quickly multiplying matrices that are "sparse"; i.e., have many entries equal to 0. Efficiency with matrices on the computer is also relevant beyond artificial intelligence problems, such as in the finite element method and weather forecasting.

14.4. Rotations revisited in $\mathbf{R}^2$. Imagine rotating $\mathbf{R}^2$ by some angle θ around the origin; let's say we do it counterclockwise. Mathematically speaking, this rotation describes a function $R_\theta : \mathbf{R}^2 \to \mathbf{R}^2$ where

$$R_\theta(\mathbf{v}) = \text{ counterclockwise rotation of } \mathbf{v} \text{ by } \theta$$

(with the understanding that $R_\theta(\mathbf{0}) = \mathbf{0}$). *The function R_θ is linear.* The reason is that rotation respects the parallelogram law for vector addition (see Figure 14.4.1) and respects scalar multiplication (changing the unit of distance before or after a rotation makes no difference), so it is linear by Theorem 14.2.1!

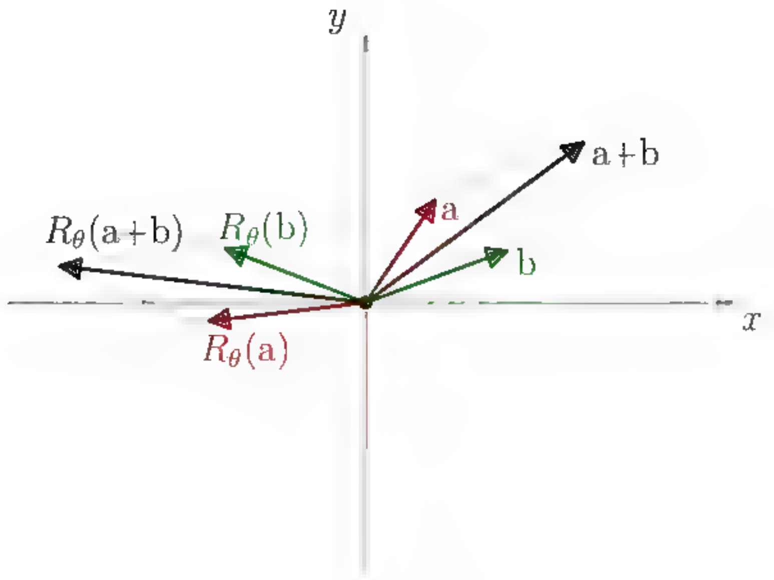


FIGURE 14.4.1. Rotation of a parallelogram is a parallelogram, so $R_\theta(\mathbf{a}+\mathbf{b}) = R_\theta(\mathbf{a}) + R_\theta(\mathbf{b})$

Note how important it was to have the characterization of linearity via properties in (14.2.1). You cannot see by pure thought that R_θ has the right formula to be a linear function (we do not yet have any formula for it!), but you can see that it has the right *properties* as in Theorem 14.2.1 and so must be linear. The same reasoning applies to rotations of $\mathbf{R}^3$ around a line through the origin!

Since R_θ is now known to be a linear function, it *must* be given by some matrix A_θ; i.e., $R_\theta(\mathbf{v}) = A_\theta\mathbf{v}$, where the right side is a matrix-vector product. What is A_θ? By (13.4.2) we have a formula:

$$A_\theta = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \text{ where } \begin{bmatrix} a \\ c \end{bmatrix} = A_\theta \begin{bmatrix} 1 \\ 0 \end{bmatrix} = R_\theta\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) \text{ and } \begin{bmatrix} b \\ d \end{bmatrix} = A_\theta \begin{bmatrix} 0 \\ 1 \end{bmatrix} = R_\theta\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right).$$

You can work out what $R_\theta\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right)$ and $R_\theta\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right)$ are by staring at rotations of $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$: the result is

$$R_\theta\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}, \quad R_\theta\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}$$

(the first by *definition* of $\cos\theta$ and $\sin\theta$, the second since $\cos(\theta+\pi/2) = -\sin\theta$, $\sin(\theta+\pi/2) = \cos\theta$).

We have shown that the matrix of counterclockwise rotation of $\mathbf{R}^2$ around the origin by θ is

$$A_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{14.4.1}$$

In words, if $\mathbf{v} \in \mathbf{R}^2$, then $A_\theta\mathbf{v}$ is obtained by rotating $\mathbf{v}$ counterclockwise by θ around the origin.

Example 14.4.1. Consider a counterclockwise rotation by 45°; i.e., $\pi/4$ radians. Since $\sin(\pi/4) = \cos(\pi/4) = 1/\sqrt{2}$ (think about an isosceles right triangle with hypotenuse of length 1), we see that

$$A_{\pi/4} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}.$$

For a *clockwise* rotation by 45°, the effect is rotating by $-45°$ counterclockwise, or in other words $-\pi/4$ radians. Since $\sin(-x) = -\sin(x)$ and $\cos(-x) = \cos(x)$, we have $\sin(-\pi/4) = -1/\sqrt{2}$ and $\cos(-\pi/4) = 1/\sqrt{2}$, so the corresponding rotation matrix is

$$A_{\;\pi/4} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}.$$

■

Example 14.4.2. If you go back to the rotations in Example 8.1.12, you can use (14.4.1) to check the correctness of the formula given there for a counterclockwise rotation by 90°: it is given by the effect of the matrix

$$A_{\pi/2} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

as asserted there. The more complicated matrix given in Example 8.1.12, namely

$$\begin{bmatrix} \sqrt{3}/2 & 1/2 \\ -1/2 & \sqrt{3}/2 \end{bmatrix},$$

is exactly A_θ with $\cos(\theta) = \sqrt{3}/2$ and $\sin(\theta) = -1/2$, which is to say $\theta = -30°$; i.e., $-\pi/6$ radians. ■

Example 14.4.3. It is a familiar fact from experience with a steering wheel (in reality or virtual reality) that composing rotations corresponds to adding angles. In symbols: $R_{\theta_1} \circ R_{\theta_2} = R_{\theta_1+\theta_2}$. Let's see that matrix multiplication gives the same conclusion: we compute the matrix product

$$\begin{aligned} \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 \\ \sin\theta_2 & \cos\theta_2 \end{bmatrix} &= \begin{bmatrix} \cos\theta_1\cos\theta_2 - \sin\theta_1\sin\theta_2 & -(\cos\theta_1\sin\theta_2 + \sin\theta_1\cos\theta_2) \\ \sin\theta_1\cos\theta_2 + \cos\theta_1\sin\theta_2 & -\sin\theta_1\sin\theta_2 + \cos\theta_1\cos\theta_2 \end{bmatrix} \\ &= \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) \end{bmatrix}. \end{aligned}$$

The essential content of this calculation is the addition laws for sine and cosine. So we see that those addition laws, which may have seemed complicated or bizarre when you encountered them when learning trigonometry, are more intuitive than you thought them to be! ■

14.5. Rotations revisited in $\mathbf{R}^3$. Finally, we return to rotations in three dimensions, focusing mainly on rotations around the x-, y- and z-axes. If you study more linear algebra then you will learn how to analyze rotations around an arbitrary line passing through the origin (as arises in computer graphics, robotics, video games, and so on).

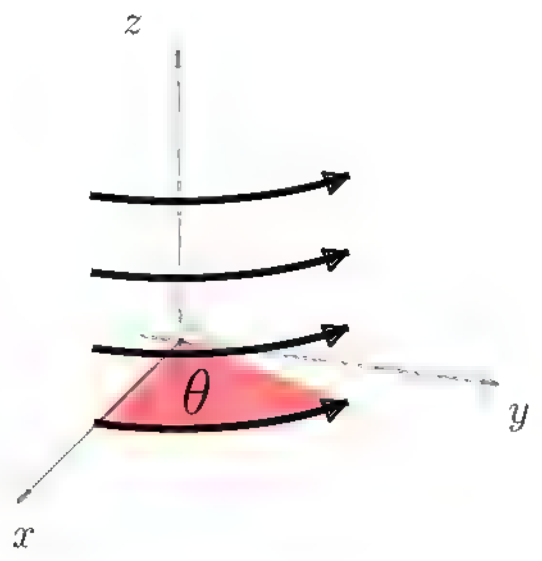


FIGURE 14.5.1. Rotation around the z-axis

What is the matrix for a rotation by an angle θ around the z-axis (as in Figure 14.5.1)? It leaves the z-coordinate unchanged (so carries $\mathbf{e}_3$ to itself) and in the x and y coordinates it does just what we worked out for $R_\theta(\mathbf{e}_1)$ and $R_\theta(\mathbf{e}_2)$ in the xy-plane in Section 14.4, so once again using the fundamental principle that *the jth column of a matrix is the effect of the linear transformation on* $\mathbf{e}_j$ we obtain

$$\text{matrix for rotation by angle } \theta \text{ counterclockwise around the } z\text{-axis is } R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Here "counterclockwise" means that you follow the right-hand rule: place your thumb in the direction of the positive z-axis, and rotate by θ in the direction that your fingers curl, as shown in Figure 14.5.1. The columns from left to right are exactly the transformation applied to $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ (check!).

Replacing the z-axis and xy-plane with the x-axis and the yz-plane respectively yields (check!):

$$\text{matrix for rotation by angle } \theta \text{ counterclockwise around the } x\text{-axis is } R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}.$$

Likewise, for rotation by an angle θ counterclockwise around the y-axis, the matrix is

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

(note the placement of $\sin(\theta)$ in the upper-right and $-\sin(\theta)$ in the lower-left!).

Remark 14.5.1. In aeronautics, the preceding three rotations have special names: $R_x(\varphi)$ is a *roll* (corresponding to a plane flying horizontally and due east tilting its wings by an angle φ relative to the ground), $R_y(\theta)$ is a *pitch* (corresponding to a plane flying due east and diverting from level flight by a vertical angle θ), and $R_z(\psi)$ is a *yaw* (corresponding to a plane flying horizontally and diverting from the eastward direction by an angle ψ; this is an ancient nautical word related to "yacht"). Alternative terms are *bank*, *elevation*, and *heading* respectively.

For a plane in flight, there is an associated triple of mutually perpendicular unit vectors: one pointing straight ahead along the line through the body of the plane, one pointing out along the left wing, and one pointing through the top of the plane. The 3×3 matrix R whose columns are these respective vectors encodes the orientation of the plane in space. This matrix can be obtained by applying pitch, yaw, and roll operations *in a specific order*: $R = R_z(\psi)R_y(\theta)R_x(\varphi)$ with $-\pi < \psi, \varphi \leq \pi$ and $-\pi/2 \leq \theta \leq \pi/2$. The angles (ψ, θ, φ) are called the *Euler angles* in the "zyx convention" or "roll-pitch-yaw" convention (they are also called *nautical angles* or *Tait-Bryan angles*); this was the convention used for Apollo 13 (beginning at 0:55 in that video; the first 8 seconds of the video are shaky), and a visualization tool is given here. There are other conventions, depending on the order of matrix multiplication; the convention used must always be specified to avoid errors.

The guidance systems on Apollo missions used Euler angles computed by mechanical gyroscopes. The case $\theta \approx \pm\pi/2$ had to be avoided because two of the three gimbals in the gyroscope would then align, after which the gyroscope cannot recover 3 degrees of freedom (a circumstance called "gimbal lock", analogous to the ambiguity in telling someone to walk due south from the North Pole). This caused serious concerns on Apollo 10 (which nearly crashed into the Moon), 11, and 13 (which nearly gyrated out of control). Mathematically, when $\theta = \pm\pi/2$ for a fixed sign, the rotation matrix

$$R_z(\psi)R_y(\pm\pi/2)R_x(\varphi) = \begin{bmatrix} 0 & \pm\sin(\varphi \mp \psi) & \pm\cos(\varphi \mp \psi) \\ 0 & \cos(\varphi \mp \psi) & -\sin(\varphi \mp \psi) \\ \pm 1 & 0 & 0 \end{bmatrix}$$

only “knows” the single angle $\varphi \mp \psi$ (for a fixed sign); it does not determine the pair of angles ψ and φ. More concretely, any change in the roll matrix $R_z(\psi)$ has the same effect as a specific change in the yaw matrix $R_x(\varphi)$ (e.g., adding 30° to ψ corresponds to adding ∓30° to φ).

A modern method in aerospace and robotics to avoid gimbal lock is to use a more robust mathematical concept, called quaternions (nicely introduced in this video); e.g., the Space Shuttle was a pioneer in the use of quaternion-based aerospace guidance systems. Unfortunately the Space Shuttle software incorporated a non-standard step (essentially flipping the order of quaternion multiplication), causing parts of the aerospace community to use quaternionic formulas that differ in crucial ways from what is used throughout mathematics, physics, standard mathematical software, and most of robotics. This created much confusion, such as for Space Shuttle docking with the International Space Station!

We are now equipped to analyze the question at the start of this chapter:

> Rotate a sphere centered at the origin first by the counterclockwise rotation R_1 around the z-axis by $-45°$, and then by the counterclockwise rotation R_2 around the y-axis by $45°$. How does this compare with composing these rotations in the opposite order?

Before we do the mathematical analysis, you should first play with a ball to try to have a sense of what is going on (looking at the effect on points marked on it with a pencil or Hagoromo chalk).

According to our preceding determination of general rotations around the coordinate axes, R_1 and R_2 have respective matrices

$$A = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 0 & 1 & 0 \\ -1/\sqrt{2} & 0 & 1/\sqrt{2} \end{bmatrix}. \tag{14.5.1}$$

We have $R_1 = T_A$ and $R_2 = T_B$, or in other words

$$R_1\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} (x/\sqrt{2}) + (y/\sqrt{2}) \\ -(x/\sqrt{2}) + (y/\sqrt{2}) \\ z \end{bmatrix}, \quad R_2\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} (x/\sqrt{2}) + (z/\sqrt{2}) \\ y \\ -(x/\sqrt{2}) + (z/\sqrt{2}) \end{bmatrix}.$$

If we first apply R_1 and then R_2, we are forming the composition $R_2 \circ R_1$ (make sure you understand why this is correct, not $R_1 \circ R_2$!). This composition is computed by the matrix product BA in the *same* order, due to the *definition* of matrix multiplication. The matrix product

$$BA = \begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 0 & 1 & 0 \\ -1/\sqrt{2} & 0 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

is the 3×3 matrix whose ij-entry is the dot product of the ith row of B and the jth column of A. For example, the $(3, 1)$-entry is the dot product

$$(-1/\sqrt{2}, 0, 1/\sqrt{2}) \cdot (1/\sqrt{2}, -1/\sqrt{2}, 0) = -1/2 + 0 + 0 = -1/2.$$

Proceeding similarly for each entry of BA, check that you get

$$BA = \begin{bmatrix} 1/2 & 1/2 & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/2 & -1/2 & 1/\sqrt{2} \end{bmatrix}.$$

How about the product AB in the other order, corresponding to the composition of rotations $T_A \circ T_B = R_1 \circ R_2$ in the opposite order? Once again using (14.3.1) and systematically computing many dot products (now for rows of A and columns of B), check that

$$AB = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 0 & 1 & 0 \\ -1/\sqrt{2} & 0 & 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 1/2 & 1/\sqrt{2} & 1/2 \\ -1/2 & 1/\sqrt{2} & -1/2 \\ -1/\sqrt{2} & 0 & 1/\sqrt{2} \end{bmatrix}. \tag{14.5.2}$$

Observe that AB is very different from BA, conveying algebraically what is also seen geometrically by rotating an actual ball in accordance with R_1 and R_2.

Example 14.5.2. The three columns of A in (14.5.1) are orthonormal, and likewise for B in (14.5.1) and for the 3×3 matrices BA and AB that we have computed. You can verify this by direct computation for each matrix (please do it yourself for at least one of them), but here is a more illuminating geometric way to see such orthonormality *without computation*.

Each of these matrices computes a linear function that preserves lengths and angles. If T is any such linear function then for the standard basis $\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ we know (as for *any* linear transformation $\mathbf{R}^3 \to \mathbf{R}^3$) that the columns of the matrix for T are $T(\mathbf{e}_1), T(\mathbf{e}_2), T(\mathbf{e}_3)$. But the effect of T preserves lengths of vectors and angles between vectors, so it carries a collection of pairwise orthogonal unit vectors to a collection of pairwise orthogonal unit vectors. It follows that if $i \neq j$ then $T(\mathbf{e}_i)$ and $T(\mathbf{e}_j)$ are perpendicular to each other (because $\mathbf{e}_i, \mathbf{e}_j$ are perpendicular) and each $T(\mathbf{e}_i)$ has length 1. This expresses that the columns of T are orthonormal. To summarize:

Any 3×3 matrix whose effect preserves lengths and angles has orthonormal columns.

In Chapter 20 we will study such matrices – and their $n \times n$ analogues – in more depth. ■

Remark 14.5.3 (online resource). The third, fourth, and fifth videos at "Essence of Linear Algebra" give dynamic visualizations of linear transformations and matrix products. Some useful online matrix calculators (for playing around and checking your work) are reshish, matrixcalc, and Wolfram.

14.6. Proofs involving linearity. Let's justify the visualization in Figure 14.1.3 and the Tiling Principle from Section 14.1, prove Theorem 14.2.1, and prove the formula (14.3.1) for a matrix product. First we justify Figure 14.1.3, showing that $\mathbf{f}(S)$ is the parallelogram with vertices shown there:

PROOF. The key to describing $\mathbf{f}(S)$ is to express S in the language of linear algebra: the points of S are $\mathbf{s} = \begin{bmatrix} x \\ y \end{bmatrix} = x \begin{bmatrix} 1 \\ 0 \end{bmatrix} + y \begin{bmatrix} 0 \\ 1 \end{bmatrix} = x\mathbf{e}_1 + y\mathbf{e}_2$ for $0 \leq x, y \leq 1$. Hence, by the Linearity Principle, the output $\mathbf{f}(S)$ of $\mathbf{f}$ applied to points $\mathbf{s} \in S$ consists of the points

$$\mathbf{f}(\mathbf{s}) = \mathbf{f}(x\mathbf{e}_1 + y\mathbf{e}_2) = x\mathbf{f}(\mathbf{e}_1) + y\mathbf{f}(\mathbf{e}_2) = x \begin{bmatrix} a \\ c \end{bmatrix} + y \begin{bmatrix} b \\ d \end{bmatrix} \tag{14.6.1}$$

for $0 \leq x, y \leq 1$. We claim that these are *exactly* the points of the parallelogram P having a vertex at $\mathbf{0}$ and edges along $\begin{bmatrix} a \\ c \end{bmatrix}$ and $\begin{bmatrix} b \\ d \end{bmatrix}$ (and so with far tip at the vector sum $\begin{bmatrix} a+b \\ c+d \end{bmatrix}$ by the parallelogram law) as drawn on the right in Figure 14.1.3.

The way that the red regions in Figure 14.1.3 sit inside the larger green regions expresses a visual interpretation of (14.6.1) and underlies why $\mathbf{f}(S) = P$. We now explain this, since it conveys the essence of linearity and how one should *always* visualize the effect of a "typical" linear transformation. **This is the heart of how algebra and geometry interact in linear algebra.**

First, we work out what $\mathbf{f}$ does to the bottom and left edges of S (confirming what is suggested by Figure 14.1.3). These edges are line segments joining $\mathbf{0}$ to $\begin{bmatrix}1\\0\end{bmatrix}$ and $\begin{bmatrix}0\\1\end{bmatrix}$ respectively. For any $\mathbf{w} \in \mathbf{R}^2$ the line segment joining $\mathbf{0}$ to $\mathbf{w}$ consists of the points $t\mathbf{w}$ for $0 \le t \le 1$, so since $\mathbf{f}(t\mathbf{w}) = t\mathbf{f}(\mathbf{w})$ by the Linearity Principle we see that the output of $\mathbf{f}$ on the segment L joining $\mathbf{0}$ to any $\mathbf{w}$ is the segment L' joining $\mathbf{0}$ to $\mathbf{f}(\mathbf{w})$. The equation $\mathbf{f}(t\mathbf{w}) = t\mathbf{f}(\mathbf{w})$ can be interpreted as saying that for the point $\mathbf{v} = t\mathbf{w} \in L$ whose distance from $\mathbf{0}$ is a fraction t of the full length of L, $\mathbf{f}(\mathbf{v}) = t\mathbf{f}(\mathbf{w})$ is the point of L' whose distance from $\mathbf{0}$ is the *same* fraction t of the full length of L' (e.g., $\mathbf{f}((1/3)\mathbf{w}) = (1/3)\mathbf{f}(\mathbf{w})$ says that the point of L which is 1/3 of the way from $\mathbf{0}$ is carried to the point of L' that is 1/3 of the way from $\mathbf{0}$).

Applying this with $\mathbf{w} = \begin{bmatrix}1\\0\end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix}0\\1\end{bmatrix}$, the output of $\mathbf{f}$ on the bottom and left edges of S is given by the edges of P joining $\mathbf{0}$ to $\mathbf{f}\left(\begin{bmatrix}1\\0\end{bmatrix}\right) = \begin{bmatrix}a\\c\end{bmatrix}$ (first column of A) and to $\mathbf{f}\left(\begin{bmatrix}0\\1\end{bmatrix}\right) = \begin{bmatrix}b\\d\end{bmatrix}$ (second column of A) respectively.

Now consider a general point $\mathbf{s} = \begin{bmatrix}x\\y\end{bmatrix} \in S$. As x, y vary between 0 and 1, $\mathbf{s}$ sweeps out all points of S, with x and y respectively keeping track of how "far" into S the point $\mathbf{s}$ is relative to the bottom and left edges of S as fractions of the entire edge lengths (which are 1, since S is the unit square). This is visualized with the red rectangle R having opposite corners $\mathbf{0}$ and $\mathbf{s}$ and side lengths that, as fractions of the full edge lengths of S, are x and y.

By (14.6.1), $\mathbf{f}(\mathbf{s}) = x\begin{bmatrix}a\\c\end{bmatrix} + y\begin{bmatrix}b\\d\end{bmatrix}$. The visualization of scalar multiplication and vector addition yields a geometric visualization of $x\begin{bmatrix}a\\c\end{bmatrix} + y\begin{bmatrix}b\\d\end{bmatrix}$ in terms of the parallelogram P whose edge lengths are generally not 1: it is the point in P whose corresponding red parallelogram Q as in Figure 14.1.3 has edges whose lengths as fractions of the entire edge lengths of P are x and y respectively. So as x, y vary between 0 and 1, these points sweep out exactly the entirety of P! This shows that $\mathbf{f}(S) = P$ (and moreover that $\mathbf{f}(R) = Q$). ☐

Now that we know $\mathbf{f}(S)$ is a parallelogram (and even the parallelogram P with vertices as in Figure 14.1.3), so it makes sense to speak of tiling $\mathbf{R}^2$ by copies of $\mathbf{f}(S)$, let's use the Linearity Principle to show that $\mathbf{f}$ transforms the tiling by S into the tiling by P:

PROOF. The key point is to express the geometric property of "tiling" in terms of algebra. Just as P consists of points $x\begin{bmatrix}a\\c\end{bmatrix} + y\begin{bmatrix}b\\d\end{bmatrix}$ for $0 \le x, y \le 1$, when we tile $\mathbf{R}^2$ by copies of P then we claim that each individual "tile" consists of points $(m + x)\begin{bmatrix}a\\c\end{bmatrix} + (n + y)\begin{bmatrix}b\\d\end{bmatrix}$ for $0 \le x, y \le 1$ and some fixed *integers* m and n. The corners of the tiling are reached by moving from $\mathbf{0}$ parallel to the edges of P in units of *full edge lengths*, so by the parallelogram law for vector addition the corners of the tiling are the points $m\begin{bmatrix}a\\c\end{bmatrix} + n\begin{bmatrix}b\\d\end{bmatrix}$ for *integers* m, n (think about this!). Each tile has a "lower left" corner $\mathbf{v} = m \begin{bmatrix} a \\ c \end{bmatrix} + n \begin{bmatrix} b \\ d \end{bmatrix}$ with integers m and n, and for varying $0 \leq x, y \leq 1$ the points

$$(m + x) \begin{bmatrix} a \\ c \end{bmatrix} + (n + y) \begin{bmatrix} b \\ d \end{bmatrix} = \left(m \begin{bmatrix} a \\ c \end{bmatrix} + n \begin{bmatrix} b \\ d \end{bmatrix} \right) + \left(x \begin{bmatrix} a \\ c \end{bmatrix} + y \begin{bmatrix} b \\ d \end{bmatrix} \right) = \mathbf{v} + \left(x \begin{bmatrix} a \\ c \end{bmatrix} + y \begin{bmatrix} b \\ d \end{bmatrix} \right)$$

sweep out an entire copy of P with $\mathbf{v}$ as its lower-left corner. This describes the tiling by copies of P.

The same works for the easier tiling by S using $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ in place of $\begin{bmatrix} a \\ c \end{bmatrix}$ and $\begin{bmatrix} b \\ d \end{bmatrix}$. By the Linearity Principle, applying $\mathbf{f}$ to a constituent $\left(m \begin{bmatrix} 1 \\ 0 \end{bmatrix} + n \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) + \left(x \begin{bmatrix} 1 \\ 0 \end{bmatrix} + y \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$ of the tiling by copies of S (for fixed integers m, n and varying $0 \leq x, y \leq 1$) yields $\mathbf{f}\left(m \begin{bmatrix} 1 \\ 0 \end{bmatrix} + n \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) + \mathbf{f}\left(x \begin{bmatrix} 1 \\ 0 \end{bmatrix} + y \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right)$. By a further application of the Linearity Principle, this is equal to

$$\left(m\mathbf{f}\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) + n\mathbf{f}\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \right) + \left(x\mathbf{f}\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) + y\mathbf{f}\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \right) = \mathbf{v} + \left(x \begin{bmatrix} a \\ c \end{bmatrix} + y \begin{bmatrix} b \\ d \end{bmatrix} \right).$$

By varying $0 \leq x, y \leq 1$ with fixed integers m and n, we have obtained a constituent of the tiling by copies of P. Hence, we have shown that $\mathbf{f}$ carries the tiling by copies of S over to the tiling by copies of P, which is exactly the Tiling Principle. ☐

Next, we prove Theorem 14.2.1.

PROOF. First, we first check that any linear function $f : \mathbf{R}^n \to \mathbf{R}^m$ satisfies (14.2.1). By definition $f(\mathbf{x}) = A\mathbf{x}$ for some $m \times n$ matrix $A = (a_{ij})$, so we have to show the vector identities

$$A(c\mathbf{x}) = c\,A\mathbf{x}, \quad A(\mathbf{x} + \mathbf{y}) = A\mathbf{x} + A\mathbf{y}.$$

We do this by comparing the ith component on both sides of each desired identity, for $1 \leq i \leq n$. The ith component of $A\mathbf{x}$ is $(A\mathbf{x})_i = a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n$, so likewise $(A(c\mathbf{x}))_i$ is given by the same formula with x_j replaced by cx_j for each j. But we can factor out c to get

$$a_{i_1}(cx_1) + a_{i2}(cx_2) + \cdots + a_{in}(cx_n) = c(a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n) = c(A\mathbf{x})_i = (c\,A\mathbf{x})_i.$$

This shows $(A(c\mathbf{x}))_i = (c\,A\mathbf{x})_i$ for all i, so $A(c\mathbf{x}) = c\,A\mathbf{x}$ as desired. The case of vector addition goes very similarly, and comes down to the identity

$$\begin{aligned} a_{i1}(x_1 + y_1) + a_{i2}(x_2 + y_2) + \cdots &= a_{i1}x_1 + a_{i1}y_1 + a_{i2}x_2 + a_{i2}y_2 + \cdots \\ &= (a_{i1}x_1 + a_{i2}x_2 + \cdots) + (a_{i1}y_1 + a_{i2}y_2 + \cdots); \end{aligned}$$

we leave it to the interested reader to fill in the details for this part.

We next explain the more interesting feature of going the other way: why any function satisfying (14.2.1) *must* be a linear function. For any vector $\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbf{R}^n$ we may write:

$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = x_1\mathbf{e}_1 + \cdots + x_n\mathbf{e}_n.$$

If $f : \mathbf{R}^n \to \mathbf{R}^m$ satisfies (14.2.1), apply f to both sides to get

$$f\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}\right) = f(x_1\mathbf{e}_1) + f(x_2\mathbf{e}_2) + \cdots + f(x_n\mathbf{e}_n) = x_1 f(\mathbf{e}_1) + x_2 f(\mathbf{e}_2) + \cdots + x_n f(\mathbf{e}_n),$$

where we used (14.2.1) for both equalities. We can rewrite this as

$$f\left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}\right) = A \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

where A is the matrix $\begin{bmatrix} f(\mathbf{e}_1) & f(\mathbf{e}_2) & \cdots & f(\mathbf{e}_n) \end{bmatrix}$ with jth column $f(\mathbf{e}_j)$. Voila, $f(\mathbf{x}) = A\mathbf{x}$ for a matrix A, so f is linear!

To show that for $\mathbf{f}(\mathbf{x}) = A\mathbf{x}$ the jth column of A is $\mathbf{f}(\mathbf{e}_j) = A\mathbf{e}_j$, we just directly compute the matrix-vector product $A\mathbf{e}_j$ to see that its ith entry is a_{ij}, so $A\mathbf{e}_j$ is the column of numbers $a_{1j}, a_{2j}, \ldots, a_{mj}$ (listed *vertically* in that order). This is the jth column of A, as desired.

Finally, suppose $\mathbf{g} : \mathbf{R}^n \to \mathbf{R}^m$ and $\mathbf{h} : \mathbf{R}^p \to \mathbf{R}^n$ are linear. To show that $\mathbf{g} \circ \mathbf{h} : \mathbf{R}^p \to \mathbf{R}^m$ is linear it is enough (by what has been shown above) to check that this interacts well with vector addition and scalar multiplication. We will do this by using that both $\mathbf{g}$ and $\mathbf{h}$ have such good behavior (as we are assuming each is linear!). For any $\mathbf{x}, \mathbf{y} \in \mathbf{R}^p$ we have

$$(\mathbf{g} \circ \mathbf{h})(\mathbf{x} + \mathbf{y}) = \mathbf{g}(\mathbf{h}(\mathbf{x} + \mathbf{y})) = \mathbf{g}(\mathbf{h}(\mathbf{x}) + \mathbf{h}(\mathbf{y})) = \mathbf{g}(\mathbf{h}(\mathbf{x})) + \mathbf{g}(\mathbf{h}(\mathbf{y})) = (\mathbf{g} \circ \mathbf{h})(\mathbf{x}) + (\mathbf{g} \circ \mathbf{h})(\mathbf{y})$$

and for any scalar c we have $(\mathbf{g} \circ \mathbf{h})(c\mathbf{x}) = \mathbf{g}(\mathbf{h}(c\mathbf{x})) = \mathbf{g}(c\,\mathbf{h}(\mathbf{x})) = c\,\mathbf{g}(\mathbf{h}(\mathbf{x})) = c\,(\mathbf{g} \circ \mathbf{h})(\mathbf{x})$. ☐

Finally, we establish (14.3.1) using the notation introduced there (such as A and B).

PROOF. We will write out the formula for $T_A \circ T_B$ to compute its matrix using some algebra, and making free use of summation notation and its properties. (If you find this derivation confusing because of the notation, please work through it yourself when both A and B are 2×2 matrices to confirm (14.3.1) by direct substitution of the formula for $T_B(\mathbf{x})$ in terms of B into the formula for $T_A(\mathbf{v})$ in terms of A to compute $T_A(T_B(\mathbf{x}))$ explicitly without using summation notation.)

Suppose $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$. Then the kth component of $T_B(\mathbf{x})$ is $\sum_{j=1}^{p} b_{kj} x_j$, so the ith component of $T_A(T_B(\mathbf{x})))$ is $\sum_{k=1}^{n} a_{ik} \sum_{j=1}^{p} b_{kj} x_j = \sum_{k=1}^{n} \sum_{j=1}^{p} a_{ik} b_{kj} x_j = \sum_{j=1}^{p} \left(\sum_{k=1}^{n} a_{ik} b_{kj}\right) x_j$, where the last step used the maneuver of swapping the order of summation. Since AB is defined to be the matrix that computes $T_A \circ T_B$, so $(AB)\mathbf{x} = (T_A \circ T_B)(\mathbf{x}) = T_A(T_B(\mathbf{x}))$, we conclude that the ith entry of $(AB)\mathbf{x}$ is given by the final double summation above. But by the meaning of matrix-vector products, $(AB)\mathbf{x}$ has ith entry $\sum_{j=1}^{p} (AB)_{ij} x_j$, where $(AB)_{ij}$ denotes the ij-entry of AB that we are seeking to compute. Equating our two expressions for the ith entry of $(AB)\mathbf{x}$ yields

$$\sum_{j=1}^{p} (AB)_{ij} x_j = \sum_{j=1}^{p} \left(\sum_{k=1}^{n} a_{ik} b_{kj}\right) x_j,$$

so equating coefficients of x_j on the two sides gives $(AB)_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$ as desired. ☐

Chapter 14 highlights (links to highlights in and next chapters)

Notation	Meaning	Location in text
T_A (for $m \times n$ matrix A)	the function $\mathbf{R}^n \to \mathbf{R}^m$ carrying $\mathbf{x} \in \mathbf{R}^n$ to $A\mathbf{x} \in \mathbf{R}^m$	Definition 14.3.1
AB (for matrices A, B)	for $m \times n$ matrix A and $n \times p$ matrix B, the $m \times p$ matrix computing effect of $T_A \circ T_B$, given by (14.3.1)	Def. 14.3.1, Thm. 14.3.2
R_θ (for angle θ)	the function $\mathbf{R}^2 \to \mathbf{R}^2$ rotating counterclockwise around $\mathbf{0}$ by the angle θ	start of Section 14.4

Concept	Meaning	Location in text
linear transformation	a function $\mathbf{R}^n \to \mathbf{R}^m$ of the form $\mathbf{f}(\mathbf{x}) = A\mathbf{x}$ for an $m \times n$ matrix A (same as "linear function"; see Prop. 13.3.8)	Definition 14.1.1
matrix multiplication	matrix that computes the effect of the composition (in a specified order) of linear transformations associated to two given matrices	Definition 14.3.1
projection matrix	for linear subspace V of $\mathbf{R}^n$, the $n \times n$ matrix computing the effective of $\mathbf{Proj}_V : \mathbf{R}^n \to \mathbf{R}^n$	Example 14.2.2
rotation matrix ($n = 2, 3$)	matrix computing effect of rotation (by some angle) of $\mathbf{R}^2$ around $\mathbf{0}$ or of $\mathbf{R}^3$ around a coordinate axis	Sections 14.4, 14.5

Result	Meaning	Location in text
characterization of linear functions via properties	a function $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ is linear precisely when it "interacts well" with vector addition and scalar multiplication	Theorem 14.2.1
projections & rotations are linear	the geometrically-defined concepts of projection to a linear subspace of $\mathbf{R}^n$ and rotation by some angle around the origin in $\mathbf{R}^2$ or a coordinate axis in $\mathbf{R}^3$ interact well with vector addition and scalar multiplication and so (as for any linear functions) are computed as the matrix-vector product against a specific matrix (found using Theorem 13.4.5!)	Ex. 14.2.2, Sec. 14.4
matrix multiplication "is" linear composition	a composition of linear functions is linear, so composition can be expressed in terms of matrices since linear functions "correspond" to matrices	Theorem 14.3.2
matrix multiplication not commutative, but commutes in the diagonal case	the order of multiplication of two matrices typically has a huge effect on their product (and on whether the product makes sense), but is "easy" in the diagonal case	Exs. 14.3.3–14.3.5, 14.3.8, 14.3.9, Prop. 14.3.7

Skill	Location in text
visualize linear $T : \mathbf{R}^2 \to \mathbf{R}^2$ via tiling and effect on unit square	Examples 14.1.2, 14.1.3
compute matrix products, be aware that order matters	Examples 14.3.3–14.3.5, 14.3.8, 14.3.9
compute matrix for rotation of $\mathbf{R}^2$ around $\mathbf{0}$ by specified angle	(14.4.1), Examples 14.4.1, 14.4.2
compute matrix for rotation of $\mathbf{R}^3$ around a coordinate axis by specified angle	Section 14.5

15. Matrix algebra

We have introduced (in Chapter 14) the notion of matrix multiplication as a means of computing composition of linear transformations, and a variety of explicit examples have been worked out (illustrating how ubiquitous it is that $AB \neq BA$ for $n \times n$ matrices A and B when $n > 1$). As the course proceeds, the importance of matrix multiplication will become ever more apparent.

However, we haven't given an especially good reason for the terminology "matrix multiplication" (other than that we chose to denote it as AB). In this chapter we will explore further properties of this operation. Its relationship with matrix addition (a new notion that we need to define, but fortunately holds no surprises at all) will demystify why we call the construction AB (which looks rather peculiar in terms of the formula for its entries) a "product". The features of matrix algebra discussed here will be used repeatedly in the chapters that follow.

By the end of this chapter you should be able to:

- rapidly compute matrix products involving a diagonal matrix;
- compute the sum of two matrices and scalar multiples of a matrix, and know the linear transformation corresponding to each and how these interact with matrix multiplication;
- appreciate that in general there is no "cancellation law" for matrix multiplication, and that matrix multiplication is sensitive to the order in which it is applied.

In the historical development of matrices, multiplication came *before* addition (see [Ha1, p. 567])! This may sound surprising, but the reason is very natural: composing rotations (as in Section 14.1) has visual meaning whereas matrix addition has only algebraic rather than geometric significance.

15.1. Some examples with matrix multiplication. Matrix multiplication in general is a complicated-looking notion, though if we think about it geometrically in terms of calculating the effect of composing linear transformations (such as rotations or shearing in some direction) then it is a natural thing to consider. Its ubiquity and significance throughout the natural sciences, economics, computer science, and data science is largely due to its connection to the composition of transformations. There are special cases when the algebraic formulas for matrix multiplication become simpler, and one of the real surprises in the theory of matrix algebra is that these special cases turn out to be highly relevant even for work with rather general matrices (as we shall see later, especially in Chapter 22 and Part V). We discuss a few of these special types of matrices now. First, we need a definition:

Definition 15.1.1. For an $n \times n$ matrix A, its *diagonal* consists of the entries a_{ii} in position (i, i) for all $1 \leq i \leq n$: this is the diagonal going from upper left to lower right, sometimes called the *main diagonal*. (The other diagonal direction, from lower left to upper right, is sometimes called the *anti-diagonal*; it plays no significant role whatsoever in linear algebra.)

More generally, for an $m \times n$ matrix B for any $m, n \geq 1$, its *diagonal* consists of its entries b_{ii}. An $m \times n$ matrix is called *diagonal* if all entries away from the diagonal vanish.

Example 15.1.2. Here are some examples of diagonal matrices:

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 4 \end{bmatrix}, \quad \begin{bmatrix} -1 & 0 \\ 0 & 5 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} -6 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 3 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{bmatrix}.$$

Here are some *non-diagonal* matrices (make sure you see why each is non-diagonal):

$$\begin{bmatrix} 2 & 1 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 4 \end{bmatrix}, \quad \begin{bmatrix} -1 & 0 \\ 6 & 5 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & -6 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & 0 \\ 3 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -2 \end{bmatrix}.$$

It's very easy to multiply two diagonal matrices: the diagonal entries just multiply! For instance:

$$\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \begin{bmatrix} a' & 0 & 0 \\ 0 & b' & 0 \\ 0 & 0 & c' \end{bmatrix} = \begin{bmatrix} aa' & 0 & 0 \\ 0 & bb' & 0 \\ 0 & 0 & cc' \end{bmatrix}, \quad \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a' & 0 & 0 & 0 & 0 \\ 0 & b' & 0 & 0 & 0 \\ 0 & 0 & c' & 0 & 0 \end{bmatrix} = \begin{bmatrix} aa' & 0 & 0 & 0 & 0 \\ 0 & bb' & 0 & 0 & 0 \\ 0 & 0 & cc' & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

■

Example 15.1.3. It is not too much harder to multiply *any* matrix by a diagonal matrix (of the appropriate size!), but the effect depends a lot on whether we multiply by the diagonal matrix on the left or right:

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 7 & 0 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} = \begin{bmatrix} 2a & 2b \\ 7c & 7d \end{bmatrix}, \quad \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 7 & 0 \end{bmatrix} = \begin{bmatrix} 2a & 7b & 0 \\ 2c & 7d & 0 \\ 2e & 7f & 0 \end{bmatrix}$$

This illustrates that for a *diagonal* matrix D, computing DA multiplies **rows** of A by the corresponding diagonal entries of D (dropping rows beyond where the diagonal of D ends, in the non-square case), and computing BD multiplies the **columns** of B by the corresponding diagonal entries of D (appending columns of 0's beyond where the diagonal of D ends, in the non-square case). Thinking about the 2×2 case, such as

$$\begin{bmatrix} 2 & 0 \\ 0 & 7 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 2a & 2b \\ 7c & 7d \end{bmatrix}, \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 7 \end{bmatrix} = \begin{bmatrix} 2a & 7b \\ 2c & 7d \end{bmatrix},$$

is the easiest way to remind yourself what happens, so *don't memorize it.* ■

The $n \times n$ diagonal matrix whose diagonal entries are all equal to 1 has a special name:

The special property of the number 1 is that $1 \times a = a = a \times 1$ for any $a \in \mathbf{R}$. There is an $n \times n$ matrix that satisfies an analogous property for multiplication of $n \times n$ matrices, and (even though typically $AB \neq BA$ when $n > 1$) it works whether you multiply on the left or right:

Definition 15.1.4. The $n \times n$ *identity matrix*, denoted I_n, is defined to be the diagonal $n \times n$ matrix

$$\mathrm{I}_n = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

whose diagonal entries are all equal to 1.

- The corresponding linear transformation $T_{\mathrm{I}_n} : \mathbf{R}^n \to \mathbf{R}^n$ gives the same output as input. In other words, $T_{\mathrm{I}_n}(\mathbf{x}) = \mathbf{x}$ for every $\mathbf{x} \in \mathbf{R}^n$ (Definition 14.3.1 defines the notation "T_A").
- For any $m \times n$ matrix A we have $\mathrm{I}_m A = A = A\mathrm{I}_n$.

It may seem that I_n is hardly deserving of any attention. But for our later work with the fundamental concepts of matrix inversion (the matrix analogue of "$1/x$", essential for studying systems of linear equations) and eigenvectors, the matrix I_n will be extremely convenient.

Remark 15.1.5. Diagonal matrices may seem too special to be of any interest. One of the great surprises in linear algebra is how ubiquitous they are. For example, in Chapter 27 we'll see that *every* $m \times n$ matrix A can be expressed as $A = RDR'$ where D is a diagonal $m \times n$ matrix (with non-negative diagonal entries) and the $m \times m$ matrix R and $n \times n$ matrix R' express "rigid motions" of $\mathbf{R}^m$ and $\mathbf{R}^n$ respectively.

In words: upon rotating the standard bases of $\mathbf{R}^m$ and $\mathbf{R}^n$ into some other orthonormal bases (encoded in R and R'), in such new reference frames the effect of A looks *diagonal* (encoded in D)! Figure B.4.1 illustrates how astonishing this is: an "ugly" A carries an ellipsoid (egg-shape) titled in a specific way exactly onto a sphere, carrying the perpendicular axes of symmetry of the ellipsoid exactly onto an orthonormal basis as shown. Such a visual meaning of RDR' always holds (this may not be apparent).

The existence of such an expression RDR' for A (called a *singular value decomposition* of A) is a deep result, but what is the practical significance? It turns out to be the most fundamental fact in the entirety of modern data science: this underlies data compression, machine learning, and all "big data" problems in all scientific fields. These matters are discussed at length in Section 27.3. Diagonal matrices arise in quantitative fields in numerous other ways via the concept of "eigenvalue" to be discussed in Part V.

Example 15.1.6. There is a nice class of matrix multiplications that can be used to manipulate rows:

$$\begin{bmatrix} 1 & 5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = \begin{bmatrix} a+5d & b+5e & c+5f \\ d & e & f \\ g & h & i \end{bmatrix} \tag{15.1.1}$$

adds 5 times the **second** row to the **first** row. In general, left multiplication by $\begin{bmatrix} 1 & m & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ with m as the $(1,2)$-entry *adds m times the second row to the first row*. Likewise, to add m times the **second** row to the **third** row, move the entry "m" to the $(3,2)$-position (leaving "0" in the (1,2)-entry); try it!

One can use a similar idea to manipulate columns by multiplying in the *other* order:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} 1 & 5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a & b+5a & c \\ d & e+5d & f \\ g & h+5g & i \end{bmatrix}$$

Analogously to (15.1.1), now we've added 5 times a *column* (which?) to another *column* (which?). ■

Example 15.1.7. A very useful class of matrices, one of which occurs on the left in (15.1.1), is the *upper triangular* matrices, such as

$$A = \begin{bmatrix} 1 & 5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} -2 & -1 & 7 & 4 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & -5 & 3 \end{bmatrix}, \quad C = \begin{bmatrix} 3 & 0 & 4 \\ 0 & -6 & 2 \\ 0 & 0 & -5 \\ 0 & 0 & 0 \end{bmatrix}.$$

To be precise, an $m \times n$ matrix U is called *upper triangular* when its only possible nonzero entries are in or above the diagonal, or equivalently all entries "below" the diagonal vanish (i.e., the ij-entry vanishes whenever $i > j$). We allow 0's in or above the diagonal too, but everything below the diagonal *must* be 0.

The formula to multiply two upper triangular matrices is complicated to write out explicitly (it isn't as clean as the diagonal case), but a nice thing happens: *if A and B are upper triangular, then AB is upper triangular too.* We saw 3×3 instances of this in Example 14.3.8. Check the following additional examples to convince yourself this holds in general:

$$\begin{bmatrix} 2 & -1 & 1 \\ 0 & -3 & 2 \\ 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} -1 & 3 & 6 & 1 \\ 0 & 4 & 5 & -3 \\ 0 & 0 & -2 & 1 \end{bmatrix} = \begin{bmatrix} -2 & 2 & 5 & 6 \\ 0 & -12 & -19 & 11 \\ 0 & 0 & -8 & 4 \end{bmatrix}, \quad \begin{bmatrix} 1 & -4 & 3 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 4 & -27 \\ 0 & 1 & -6 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\begin{bmatrix} -1 & 3 & 6 & 1 \\ 0 & 4 & 5 & -3 \\ 0 & 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} 6 & 2 & 4 \\ 0 & 1 & 2 \\ 0 & 0 & -4 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -6 & 1 & -22 \\ 0 & 4 & -12 \\ 0 & 0 & 8 \end{bmatrix}.$$

Similarly, we can define *lower triangular matrices*, such as

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 5 & 1 & 0 \\ 0 & 4 & -3 \end{bmatrix}, \quad B = \begin{bmatrix} -2 & 0 & 0 & 0 \\ 6 & 1 & 0 & 0 \\ 4 & -3 & 5 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 3 & 0 & 0 \\ 2 & -6 & 0 \\ 4 & 7 & -5 \\ 9 & -2 & -1 \end{bmatrix}.$$

To be precise, an $m \times n$ matrix L is called *lower triangular* when its only possible nonzero entries are in or below the diagonal, or equivalently all entries "above" the diagonal vanish (i.e., the ij-entry vanishes whenever $i < j$). We allow 0's in or below the diagonal too, but everything above the diagonal *must* be 0. The product of two lower triangular matrices is always a lower triangular matrix, much like the upper triangular case considered above. ■

Remark 15.1.8. You might think that upper and lower triangular matrices are too special to be of any real interest for work with general matrices. But amazingly, it turns out that "most" $m \times n$ matrices A can be written in the form $A = LU$ where L is $m \times p$ lower triangular and U is $p \times n$ upper triangular for $p = \min(m, n)$ the smaller among m and n (so if $m = n$ then $p = m = n$)! It is mysterious to us at this stage how to find such L and U when given A. In Chapter 22 we will discuss the application of such L and U to solving systems of m linear equations in n unknowns, and for computing "matrix inverses", and indicate how such L and U are computed from A.

15.2. Addition and scalar multiplication for matrices. Before we state the general properties of matrix multiplication, we need to address *addition* and *scalar multiplication* for matrices, topics that are fortunately entirely unsurprising: if A and B are $m \times n$ matrices and c is a scalar, then:

(i) the *sum* $A + B$ is defined to be the $m \times n$ matrix obtained by adding corresponding entries in A and B (i.e., its ij-entry is $a_{ij} + b_{ij}$); **the matrix sum $A + B$ only makes sense when A and B are both $m \times n$**,

(ii) the *scalar multiple* cA is defined to be the $m \times n$ matrix obtained by multiplying every entry of A by c (i.e., its ij-entry is ca_{ij}).

Example 15.2.1. Here are some matrix sums:

$$\begin{bmatrix} -1 & 2 & 1 \\ 0 & 7 & 4 \\ 5 & -3 & 6 \end{bmatrix} + \begin{bmatrix} 3 & 8 & 2 \\ 6 & -4 & 1 \\ 2 & 7 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 10 & 3 \\ 6 & 3 & 5 \\ 7 & 4 & 7 \end{bmatrix},$$

$$\begin{bmatrix} 1 & -1 & 6 \\ 2 & 3 & -2 \end{bmatrix} + \begin{bmatrix} -4 & 5 & -3 \\ -2 & 1 & 1 \end{bmatrix} = \begin{bmatrix} -3 & 4 & 3 \\ 0 & 4 & -1 \end{bmatrix}, \quad \begin{bmatrix} 7 & -9 \\ 4 & 5 \end{bmatrix} + \begin{bmatrix} -7 & 9 \\ -4 & -5 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

■

The matrix sums in Example 15.2.1 come out the same way if we swap the order of the matrices on the left side of each equality, ultimately because the termwise sum of numbers $a_{ij} + b_{ij}$ is the same as $b_{ij} + a_{ij}$. In this way, one gets straight from the definition that $A + B = B + A$ for any two $m \times n$ matrices A and B for any m and n due to the analogous property for numbers. Nothing surprising happens with matrix addition!

Example 15.2.2. Here are some scalar multiplications:

$$5\begin{bmatrix}3 & 8 & 2\\ 6 & -4 & 1\\ 2 & 7 & 1\end{bmatrix} = \begin{bmatrix}15 & 40 & 10\\ 30 & -20 & 5\\ 10 & 35 & 5\end{bmatrix},\ (-2)\begin{bmatrix}7 & -9\\ 4 & 5\end{bmatrix} = \begin{bmatrix}-14 & 18\\ -8 & -10\end{bmatrix},\ (-1)\begin{bmatrix}-4 & 5 & -3\\ -2 & 1 & 1\end{bmatrix} = \begin{bmatrix}4 & -5 & 3\\ 2 & -1 & -1\end{bmatrix}.$$

■

Matrix addition and scalar multiplication of matrices have meaning in terms of linear transformations:

$$T_{A+B}(\mathbf{x}) = T_A(\mathbf{x}) + T_B(\mathbf{x}),\ \ T_{cA}(\mathbf{x}) = c\,T_A(\mathbf{x})$$

(using "T_A" notation from Definition 14.3.1); i.e., these respectively correspond to the pointwise sum of linear transformations $T_A + T_B$ and the pointwise c-multiple of a linear transformation. This expresses concrete identities for matrix-vector products with respect to addition and scalar multiplication of matrices: $(A+B)\mathbf{x} = A\mathbf{x} + B\mathbf{x}$ and $(cA)\mathbf{x} = c(A\mathbf{x})$ (identities that one can check in the 2×2 case by writing them out; those who are interested in the general case of each can work that out similarly). In summary:

Consider a scalar c and two $m \times n$ matrices

$$A = \begin{bmatrix}a_{11} & a_{12} & \dots & a_{1n}\\ a_{21} & a_{22} & \dots & a_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ a_{m1} & a_{m2} & \dots & a_{mn}\end{bmatrix} \quad B = \begin{bmatrix}b_{11} & b_{12} & \dots & b_{1n}\\ b_{21} & b_{22} & \dots & b_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ b_{m1} & b_{m2} & \dots & b_{mn}\end{bmatrix}$$

Definition 15.2.3. The *matrix sum* $A + B$ is defined to be the $m \times n$ matrix

$$A + B = \begin{bmatrix}a_{11}+b_{11} & a_{12}+b_{12} & \dots & a_{1n}+b_{1n}\\ a_{21}+b_{21} & a_{22}+b_{22} & \dots & a_{2n}+b_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ a_{m1}+b_{m1} & a_{m2}+b_{m2} & \dots & a_{mn}+b_{mn}\end{bmatrix}$$

and the *scalar multiple* cA is defined to be the $m \times n$ matrix

$$cA = \begin{bmatrix}ca_{11} & ca_{12} & \dots & ca_{1n}\\ ca_{21} & ca_{22} & \dots & ca_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ ca_{m1} & ca_{m2} & \dots & ca_{mn}\end{bmatrix}$$

Proposition 15.2.4. The linear transformations T_{A+B} and T_{cA} for the matrix sum and the scalar multiple respectively satisfy $T_{A+B}(\mathbf{x}) = T_A(\mathbf{x}) + T_B(\mathbf{x})$ and $T_{cA}(\mathbf{x}) = c\,T_A(\mathbf{x})$ for all $\mathbf{x} \in \mathbf{R}^n$.

Familiar-looking rules for the interaction of matrix addition and scalar multiplication of matrices are valid and involve no surprises, such as $c(A+B) = cA + cB$ and $(c_1c_2)A = c_1(c_2A)$, since these express analogous familiar properties for numbers at the level of each individual matrix entry. Matters become more interesting (and more useful) when matrix multiplication is brought in, so we now turn to that.

15.3. First properties of matrix algebra. Part of this course is familiarizing yourself with what works – and what doesn't – with matrix multiplication. Here are some basic unsurprising properties, similar to the case of ordinary numbers (in Chapter 18 we'll encounter more advanced ones).

Important basic properties of matrix multiplication:

(MM1) It recovers matrix-vector multiplication: if A is an $m \times n$ matrix, and $\mathbf{x} \in \mathbf{R}^n$ is thought of as an $n \times 1$ matrix, the matrix-matrix product $A\mathbf{x}$ is the same as the matrix-vector product.

(MM2) $A(B+C) = AB + AC$ and $(A' + B')C' = A'C' + B'C'$. (These "distributive laws" are the reason we call it matrix multiplication.)

(MM3) $A(BC) = (AB)C$, and $A(cB) = (cA)B = c(AB)$ for any scalar c. In particular, taking C to be an $m \times 1$ matrix that is a column vector $\mathbf{v}$ by another name, $A(B\mathbf{v}) = (AB)\mathbf{v}$.

(MM4) If A is an $m \times n$ matrix, then $\mathrm{I}_m A = A = A\mathrm{I}_n$, where I_m is the $m \times m$ identity matrix and I_n is the $n \times n$ identity matrix.

In Section 15.4 we give clean proofs of these properties for those who are interested, especially an elegant proof of (MM2) and (MM3) by arguments with linear transformations instead of messy algebra with explicit formulas. Those who are interested can check that $A(c\mathrm{I}_n) = cA = (c\mathrm{I}_m)A$; i.e., cA is the product in either order of A against the diagonal matrix whose diagonal entries are all equal to c.

Example 15.3.1. Since all of the properties listed above express analogues of familiar properties for numbers, so there are no surprises, we content ourselves with illustrating these with 2×2 matrices. Let's choose

$$A = \begin{bmatrix} 3 & -1 \\ 4 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} -5 & 7 \\ 4 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 8 & 3 \\ 7 & -3 \end{bmatrix}.$$

Then

$$A(BC) = \begin{bmatrix} 3 & -1 \\ 4 & 2 \end{bmatrix} \left(\begin{bmatrix} -5 & 7 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} 8 & 3 \\ 7 & -3 \end{bmatrix} \right) = \begin{bmatrix} 3 & -1 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} 9 & -36 \\ 39 & 9 \end{bmatrix} = \begin{bmatrix} -12 & -117 \\ 114 & -126 \end{bmatrix}$$

and

$$(AB)C = \left(\begin{bmatrix} 3 & -1 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} -5 & 7 \\ 4 & 1 \end{bmatrix} \right) \begin{bmatrix} 8 & 3 \\ 7 & -3 \end{bmatrix} = \begin{bmatrix} -19 & 20 \\ -12 & 30 \end{bmatrix} \begin{bmatrix} 8 & 3 \\ 7 & -3 \end{bmatrix} = \begin{bmatrix} -12 & -117 \\ 114 & -126 \end{bmatrix},$$

so in this case $A(BC) = (AB)C$ as in (MM3).

The distributive laws $A(B+C) = AB + AC$ and $(A+B)C = AC + BC$ in (MM2) work out equally well by direct calculation. Let's discuss the first one, and leave it to the reader to carry out the analogous calculation to verify the second one in this case. The sum $B + C$ is equal to $\begin{bmatrix} 3 & 10 \\ 11 & -2 \end{bmatrix}$, so

$$A(B+C) = \begin{bmatrix} 3 & -1 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} 3 & 10 \\ 11 & -2 \end{bmatrix} = \begin{bmatrix} -2 & 32 \\ 34 & 36 \end{bmatrix}.$$

The products AB and AC are

$$AB = \begin{bmatrix} 3 & -1 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} -5 & 7 \\ 4 & 1 \end{bmatrix} = \begin{bmatrix} -19 & 20 \\ -12 & 30 \end{bmatrix}$$

and

$$AC = \begin{bmatrix} 3 & -1 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} 8 & 3 \\ 7 & -3 \end{bmatrix} = \begin{bmatrix} 17 & 12 \\ 46 & 6 \end{bmatrix},$$

so

$$AB + AC = \begin{bmatrix} -2 & 32 \\ 34 & 36 \end{bmatrix},$$

and that in turn is exactly what we calculated for $A(B+C)$ in this case.

Finally, for (MM1) and (MM4) we do direct calculations for the same A. If $\mathbf{x} \in \mathbf{R}^2$ then the matrix-matrix product $A\mathbf{x}$ is equal to

$$\begin{bmatrix} 3 & -1 \\ 4 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3x_1 - x_2 \\ 4x_1 + 2x_3 \end{bmatrix},$$

but the right side viewed as a vector is exactly the matrix-vector product $A\mathbf{x}$. That establishes (MM1) for this specific A, and for (MM4) we multiply matrices explicitly:

$$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}3 & -1\\ 4 & 2\end{bmatrix} = \begin{bmatrix}3 & -1\\ 4 & 2\end{bmatrix} = \begin{bmatrix}3 & -1\\ 4 & 2\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}.$$

■

Important properties which indicate what can "go wrong":

- You can *only* make sense of AB if the number of columns in A matches the number of rows in B. Otherwise it simply doesn't make sense. If you find yourself trying to compute AB when this condition on rows and columns doesn't hold then somewhere you have made a mistake.
- **In particular, just because AB is defined doesn't mean that BA is defined! Even if AB and BA are both defined, AB does not have to equal BA** (see Sections 14.3 and 14.5)!
- If $AB = AC$ you cannot in general "cancel" A to conclude $B = C$ when A is not the zero matrix; see Example 14.3.4. (We will study conditions on A under which a cancellation law does work when we discuss *matrix inverses* in Chapter 18.)

Example 15.3.2. If A is a 3×2 matrix and B is a 2×2 matrix then AB makes sense (as a 3×2 matrix) but BA makes no sense (since B has 2 columns and A has 3 rows). For example, if

$$A = \begin{bmatrix}4 & 3\\ -1 & 0\\ -5 & 6\end{bmatrix},\quad B = \begin{bmatrix}8 & -4\\ 0 & 1\end{bmatrix}$$

then it makes sense to form the product AB and compute it to be

$$AB = \begin{bmatrix}4 & 3\\ -1 & 0\\ -5 & 6\end{bmatrix}\begin{bmatrix}8 & -4\\ 0 & 1\end{bmatrix} = \begin{bmatrix}32 & -13\\ -8 & 4\\ -40 & 26\end{bmatrix}$$

whereas the matrix product $BA = \begin{bmatrix}8 & -4\\ 0 & 1\end{bmatrix}\begin{bmatrix}4 & 3\\ -1 & 0\\ -5 & 6\end{bmatrix}$ makes no sense at all. ■

Example 15.3.3. The distributive law for scalars explains "$(a+b)(c+d) = ac+ad+bc+bd$" (indeed, $(a+b)(c+d) = a(c+d)+b(c+d) = ac+ad+bc+bd$), and the same works for matrices for the *same reason* provided that we are careful about the order of matrix multiplication:

$$(A+B)(C+D) = A(C+D) + B(C+D) = AC + AD + BC + BD.$$

As a special case with $n \times n$ matrices, $(A+B)^2 = (A+B)(A+B) = A^2 + AB + BA + B^2$. This is *not* $A^2 + 2AB + B^2$ except when $BA = AB$, so there is no "binomial theorem" for computing $(A+B)^m$ with general $n \times n$ matrices when $n > 1$. Another special case occurs when $A = \mathrm{I}_n$ and $C = \mathrm{I}_n$:

$$(\mathrm{I}_n + B)(\mathrm{I}_n + D) = \mathrm{I}_n^2 + \mathrm{I}_n D + B\mathrm{I}_n + BD = \mathrm{I}_n + D + B + BD. \tag{15.3.1}$$

■

15.4. Proofs of properties of matrix multiplication. In this section we use the relationship between matrices and linear transformations to give a clean proof of the "associative law" $A(BC) = (AB)C$ and the "distributive laws" $A(B+C) = AB + AC$ and $(A'+B')C' = A'C' + B'C'$ for matrix multiplication, and also establish the other basic properties that for any $m \times n$ matrix A we have $\mathrm{I}_m A = A = A\mathrm{I}_n$ and the matrix-matrix product $A\mathbf{x}$ for any $\mathbf{x} \in \mathbf{R}^n$ viewed as an $n \times 1$ matrix coincides with the matrix-vector product denoted the same way.

Associative property. Let's begin with the property $A(BC) = (AB)C$. Firstly, BC is the matrix of the linear transformation $T_B \circ T_C$. Therefore, $A(BC)$ is the matrix of the linear transformation $T_A \circ (T_B \circ T_C)$. This sends a vector $\mathbf{x}$ to

$$T_A(T_B \circ T_C(\mathbf{x})) = T_A(T_B(T_C(\mathbf{x}))).$$

Going the other way, AB is the matrix of the linear transformation $T_A \circ T_B$. Therefore $(AB)C$ is the matrix of the linear transformation $(T_A \circ T_B) \circ T_C$. This transformation sends an input $\mathbf{x}$ to

$$(T_A \circ T_B)(T_C(\mathbf{x})) = T_A(T_B(T_C(\mathbf{x}))).$$

So $(AB)C$ and $A(BC)$, considered as linear transformations, are the same: they take an input $\mathbf{x}$ to the output $T_A(T_B(T_C(\mathbf{x})))$. Thus, $(AB)C = A(BC)$ since for any linear transformation $T(\mathbf{x}) = M\mathbf{x}$ with a matrix M, T determines M via the outputs $T(\mathbf{e}_j)$ (these are the columns of M).

Distributive properties. We treat the property $A(B + C) = AB + AC$, and the interested reader can adapt it to similarly establish $(A' + B')C' = A'C' + B'C'$. As with associativity, it is the same to show that the associated linear transformations $T_{A(B+C)}$ and T_{AB+AC} coincide. Evaluating on a vector $\mathbf{x}$, we have

$$T_{A(B+C)}(\mathbf{x}) = T_A(T_{B+C}(\mathbf{x})) = T_A(T_B(\mathbf{x}) + T_C(\mathbf{x})) = T_A(T_B(\mathbf{x})) + T_A(T_C(\mathbf{x})),$$

where the final equality uses that $T_A(\mathbf{v}_1 + \mathbf{v}_2) = T_A(\mathbf{v}_1) + T_A(\mathbf{v}_2)$ (applied with $\mathbf{v}_1 = T_B(\mathbf{x})$, $\mathbf{v}_2 = T_C(\mathbf{x})$). Likewise,

$$T_{AB+AC}(\mathbf{x}) = T_{AB}(\mathbf{x}) + T_{AC}(\mathbf{x}) = T_A(T_B(\mathbf{x})) + T_A(T_C(\mathbf{x})),$$

where the final equality uses the definition of matrix multiplication in terms of computing a composition of linear functions. These calculations show that for all vectors $\mathbf{x}$ the outputs $T_{A(B+C)}(\mathbf{x})$ and $T_{AB+AC}(\mathbf{x})$ are equal to the same thing and so are equal to each other. Hence, we get the equality of linear functions $T_{A(B+C)} = T_{AB+AC}$, as required.

Further properties. For an $m \times n$ matrix A we shall compute the matrix products $\mathrm{I}_m A$, $A\mathrm{I}_n$, and $A\mathbf{x}$ for any $\mathbf{x} \in \mathbf{R}^n$ viewed as an $n \times 1$ matrix. For these the explicit matrix multiplication is quite clean (please try), but the viewpoint of linear transformations is quite slick: for any $\mathbf{x} \in \mathbf{R}^n$

$$T_{\mathrm{I}_m A}(\mathbf{x}) = T_{\mathrm{I}_m}(T_A(\mathbf{x})) = T_A(\mathbf{x})$$

since $T_{\mathrm{I}_m} : \mathbf{R}^m \to \mathbf{R}^m$ carries each vector to itself, so $T_{\mathrm{I}_m A} = T_A$ as linear transformation (as they take each input $\mathbf{x}$ to the same output). This forces $\mathrm{I}_m A = A$. A similar argument shows that $A\mathrm{I}_n = A$.

Lastly, the matrix-matrix product $A\mathbf{x}$ for $\mathbf{x} \in \mathbf{R}^n$ is

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix},$$

and this is the $m \times 1$ matrix whose ith entry is

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n;$$

that in turn is the ith entry of the matrix-vector product $A\mathbf{x}$, so we are done.

Chapter 15 highlights (links to highlights in and next chapters)

Notation	**Meaning**	**Location in text**
I_n	the $n \times n$ identity matrix	Definition 15.1.4
$A+B$ (for $m\times n$ matrices A, B)	entrywise sum of matrices of same size, corresponds to pointwise sum of associated linear functions	(i) at start of Sec. 15.2
$c\,A$ (for scalar c, matrix A)	entrywise multiplication by c, corresponds to pointwise multiplication by c against output of associated linear function	(ii) at start of Sec. 15.2

Concept	**Meaning**	**Location in text**
diagonal of $m \times n$ matrix	the list of ii-entries for all i	Definition 15.1.1
diagonal $m \times n$ matrix	matrix whose non-diagonal entries all equal 0	Definition 15.1.1
upper (resp. lower) triangular $m \times n$ matrix	matrix whose only possible nonzero entries are in or above (resp. in or below) the diagonal	Example 15.1.7
matrix addition	for two $m \times n$ matrices A and B, it is the $m \times n$ matrix obtained by adding corresponding entries	Def. 15.2.3
scalar mult. against a matrix	for $m\times n$ matrix A and scalar c, it is the $m\times n$ matrix obtained by multiplying all entries by c	Definition 15.2.3

Result	**Meaning**	**Location in text**
identity matrix has no effect under matrix multiplication	for $m \times n$ matrix A, we have $\mathrm{I}_m A = A = A\,\mathrm{I}_n$	Definition 15.1.4
product of upper triangular matrices is upper triangular, similarly for lower triangular	if U_1 and U_2 are upper triangular and $U_1 U_2$ makes sense then it is upper triangular, and similarly in the lower triangular case	Example 15.1.7
$T_{A+B} = T_A + T_B$	for $m \times n$ matrices A and B and any $\mathbf{x} \in \mathbf{R}^n$, $T_{A+B}(\mathbf{x}) = T_A(\mathbf{x}) + T_B(\mathbf{x})$ in $\mathbf{R}^m$	Proposition 15.2.4
$T_{cA} = c\,T_A$	for $m\times n$ matrix A and scalar c and any $\mathbf{x} \in \mathbf{R}^n$, $T_{cA}(\mathbf{x}) = c\,T_A(\mathbf{x})$ in $\mathbf{R}^m$	Proposition 15.2.4
good properties of matrix multiplication	matrix mult. is associative & distributes on both sides over matrix addition, and interacts well with scalar mult. and matrix-vector products	(MM1)–(MM4) at start of Sec. 15.3

Skill	**Location in text**
recognize by inspection if an $m \times n$ matrix is diagonal or upper triangular or lower triangular	Example 15.1.2
know (or be able to quickly figure out) the pattern for the outcome of left or right multiplication by a diagonal $m \times n$ matrix	Examples 15.1.2, 15.1.3
add $m \times n$ matrices and multiply them by scalars	Examples 15.2.1, 15.2.2
be aware of the subtleties of matrix multiplication	box just above Example 15.3.2

16. Applications of matrix algebra: population dynamics, PageRank, and gambling

Matrix multiplication looks complicated, but its virtue is that it can be used to encode complicated phenomena (and this is *useful* – not just notation – precisely because matrix multiplication has *very nice properties*, as we will explore during the rest of this book). In this chapter we use matrix multiplication to model a class of situations in which the evolution of a system is governed at each stage by (i) knowledge of the previous stage (and no previous ones, so it is a "memoryless" process), (ii) a matrix of probabilities for transitioning among different possibilities. Such a process is called a *Markov*[16] *chain*.

The remarkable facts are that the *long-term behavior* of such systems reveals rather surprising behavior (which we will understand via ideas and methods from matrix algebra) and that these systems are ubiquitous in real life. The sections in this chapter provide a wide range of illustrations of the power and utility of Markov chains, in areas as diverse as population dynamics, computer science, and gambling (and there is a truly vast array of other applications that we do not have time or space to develop in statistical mechanics, automated speech recognition, financial forecasting, computer-based navigation, and so on).

By the end of this chapter, you should be able to:

- translate a written description of a "Markov chain" system into a mathematical description using matrix powers;
- appreciate the utility of matrix algebra to analyze long-term behavior that is hopeless to determine by direct means and involves unexpected patterns.

There is almost no new mathematics in this chapter. Rather you should focus on the skill of translating, back and forth, between verbal and mathematical descriptions of the situation. This is a genuine skill: handling "word problems" is one of the primary things you should try to get out of a math course. Computers can solve a lot of math problems for you, but *you have to know what to ask them.*

Later on (in Chapter 27) we will return to the topics in this chapter and add some serious new mathematics: we will show how notions to be introduced in Chapters 23 and 24 can be used in these situations to understand more deeply the phenomena explored in this chapter.

16.1. Bird migration and PageRank. We begin with an example concerning the migration of birds. (The details of this example are not realistic, but the underlying idea is relevant in population dynamics, evolutionary biology, public health crises, economic simulations for employment and finance, etc.) Imagine there are three islands inhabited by birds. An ornithologist studying the birds observes the following migration pattern:

- All the birds on island A migrate to another island each year. One half migrates to island B, the other half migrates to island C.
- Similarly, all the birds on island B migrate to another island each year. One half migrates to island A, and the other half migrates to island C.
- Finally, one-third of the birds on island C stay in place each year, one third migrate to island A, and one third migrate to island B.

This verbal description is illustrated visually in Figure 16.1.1. Initially, there are 10000 birds on each island. What happens after 20 years (assuming no births or deaths)?

---

[16] A.A. Markov (1856-1922) was a Russian mathematician whose introduction of the initial ideas around the concept of what were later called Markov chains was partly inspired by his interest in poetry. He was an anti-tsarist political activist; when the House of Romanov celebrated its 300th anniversary in 1913, Markov organized a counter-celebration of the 200th anniversary of the publication of a book containing the first proof of (a special case of) the Law of Large Numbers.

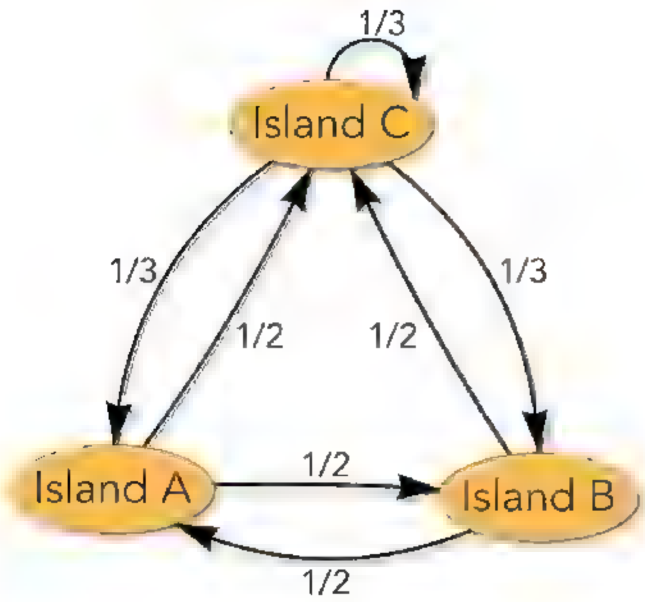


FIGURE 16.1.1. Annual bird migration pattern among the three islands

Denote the number of birds on islands A, B, C after k years as A_k, B_k, C_k respectively (so A_0, B_0, C_0 are all equal to 10000, corresponding to the initial population of 10000 birds on each island). Translating the above text into equations, it says (check!) that the populations after $k+1$ years are given as follows:

$$A_{k+1} = \frac{1}{2}B_k + \frac{1}{3}C_k, \quad B_{k+1} = \frac{1}{2}A_k + \frac{1}{3}C_k, \quad C_{k+1} = \frac{1}{2}A_k + \frac{1}{2}B_k + \frac{1}{3}C_k$$

(e.g., $(1/2)B_k$ contributes to both A_{k+1} and C_{k+1}, and $(1/3)C_k$ contributes to $A_{k+1}, B_{k+1}, C_{k+1}$). We can express this system of 3 equations in terms of a matrix-vector product (please check this!):

$$\begin{bmatrix} A_{k+1} \\ B_{k+1} \\ C_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 1/2 & 1/3 \\ 1/2 & 0 & 1/3 \\ 1/2 & 1/2 & 1/3 \end{bmatrix} \begin{bmatrix} A_k \\ B_k \\ C_k \end{bmatrix}. \tag{16.1.1}$$

Let us call the 3×3 matrix here M, and write $\mathbf{n}_k$ to denote $\begin{bmatrix} A_k \\ B_k \\ C_k \end{bmatrix}$ (e.g., the initial 3-vector $\mathbf{n}_0$ has all entries equal to 10000), so (16.1.1) says $\mathbf{n}_{k+1} = M\mathbf{n}_k$ for $k = 0, 1, 2, 3, \ldots$. The entries in each column of M are non-negative and sum to 1; any square matrix with such properties is called a *Markov matrix*.[17]

Now matrix multiplication enters: feeding the relation "$\mathbf{n}_{k+1} = M\mathbf{n}_k$" into itself repeatedly yields

$$\mathbf{n}_{20} = M\mathbf{n}_{19} = M(M\mathbf{n}_{18}) = M^2\mathbf{n}_{18} = M^2(M\mathbf{n}_{17}) = M^3\mathbf{n}_{17} = \cdots = M^{20}\mathbf{n}_0 = M^{20}\begin{bmatrix} 10000 \\ 10000 \\ 10000 \end{bmatrix}.$$

Fortunately, a computer can rapidly compute M^{20}. (We'll discuss *how* it does so in Section 24.4; there are much better ways than multiplying the matrix by itself 20 times.) Your computer says

$$M^{20} \approx \begin{bmatrix} 0.286 & 0.286 & 0.286 \\ 0.286 & 0.286 & 0.286 \\ 0.429 & 0.429 & 0.429 \end{bmatrix}, \quad \text{and so} \quad M^{20}\begin{bmatrix} 10000 \\ 10000 \\ 10000 \end{bmatrix} \approx \begin{bmatrix} 8570 \\ 8570 \\ 12860 \end{bmatrix}.$$

Hence, there are approximately 8600 birds on each of islands A and B after 20 years. Raising a matrix to a power has modeled a chain of events feeding into itself, repeating the same behavior over and over.

---

[17] These are also called *stochastic matrices*. Beware that in many references, the definition is that the sum along each *row* – rather than the sum along each column – is equal to 1; the top of the Wikipedia page for "stochastic matrix" addresses these various definitions. The column-sum definition is more convenient for our purposes, so we stick with that in this book.

Remark 16.1.1 (online resource). The websites Wolfram and reshish quickly compute matrix powers.

Remark 16.1.2 (Relation with PageRank). The columns of M^{20} are *all the same* (to the accuracy shown)! More amazingly, for all Markov matrices $\mathcal{M}$ with *positive* entries as well as certain Markov matrices (such as M above) with some entries equal to 0, the powers $\mathcal{M}^r$ for **all** sufficiently big r have all columns essentially equal to a single vector $\mathbf{v}_{\mathcal{M}}$ that is **independent** of r. The explanation rests on a result (see Section D.2) that underlies both *Google's PageRank algorithm* (read Appendix D) and the technique called "Markov Chain Monte Carlo" that is used everywhere for efficient sampling of probabilistic scenarios on a large but finite set of outcomes (applications arise in computational physics [**Kr**], computational chemistry [**FS**], social science [**Gill**, Ch. 8-13], and detecting partisan gerrymandering [**Du**, Sec. 4]).

In the above bird example, for all $r \geq 20$ it turns out that $M^r \approx M^{20}$ to high accuracy. Thus, the populations on each island *stabilize* after not so many years have passed: individual birds keep moving around, but eventually the *total population* on each island remains the same in all subsequent years. The equality of all columns has a remarkable consequence: the proportion of the total population that winds up on each island in the final "steady state" is always the corresponding entry in the common column vector (e.g., 42.9% on island C), so it is *independent* of the way the 30000 birds were initially distributed across the three islands! (The reason is a bit of algebra: if A is an $n \times n$ matrix with all columns nearly equal to a common $\mathbf{v}$ – such as M^r with big r – and $\mathbf{c}$ is any n-vector whatsoever whose entries sum to 1 – such as an "initial population vector" divided by the total population – then $A\mathbf{c} \approx c_1\mathbf{v} + \cdots + c_n\mathbf{v} = (c_1 + \cdots + c_n)\mathbf{v} = \mathbf{v}$ *regardless* of $\mathbf{c}$.) This type of astonishing conclusion arises in genetics, economics, industrial design, etc., and is crucial to the functioning of PageRank (as is discussed in Appendix D).

The lesson from this is: matrix powers encode useful (and sometimes rather surprising!) information about chains of events.

Remark 16.1.3. We have discussed near the start of Remark 16.1.2 that big powers of a Markov matrix M stabilize when the entries are either all *positive* or when the matrix satisfies some auxiliary condition which we did not formulate. The case when some entries of M equal 0 is addressed near the start of Section D.2, but let's illustrate here that things can indeed go awry in some such cases (so an extra hypothesis really is required to ensure the stabilizing of M^r for big r when M has some entries equal to 0). Consider

$$M = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

In this case we have $M^2 = \mathrm{I}_2$, so $M^3 = MM^2 = M\mathrm{I}_2 = M$, $M^4 = MM^3 = MM = M^2 = \mathrm{I}_2$, $M^5 = MM^4 = M\mathrm{I}_2 = M$, and so on: the powers of M alternate between M and I_2, with $M^{\mathrm{odd}} = M$ and $M^{\mathrm{even}} = \mathrm{I}_2$. So in this case there is no stabilizing behavior for M^r as r grows!

16.2. Gambler's ruin and genetic evolution. Alice with \$2 in her pocket and Bob with \$3 in his pocket play a game whose rules go as follows. They flip a biased coin which has 1/3 chance to be heads and 2/3 chance to be tails . If it comes up heads, Alice gives Bob a dollar; if it comes up tails, Bob gives Alice a dollar. Eventually one of them runs out of money, at which point the game ends. What happens after the first turn? There is a 1/3 chance that Alice will end up with \$1, and a 2/3 chance that Alice will end up with \$3. Since Alice has not yet reached \$5 (so Bob still has some money) nor \$0 (so Alice still has some money), the game goes on for at least one more turn.

What are the odds of various possibilities for what may happen after two turns? Or more turns? This is complicated, since the outcome at each turn depends on the previous turn, which itself is a probabilistic situation. Matrix algebra is the right tool to analyze the long-term question: what are the odds that eventually Alice reaches \$5 (i.e., she wins), or the odds that eventually Alice reaches \$0 ("gambler's ruin")? You might reason without doing any work that Bob is a fool to play this game, since Alice's value goes up 2/3 of the time, but the 1/3 versus 2/3 bias is just a magnified version of the "house edge" that every gambler in a casino hopes to overcome. Regardless, Alice does have some positive chance of losing (likely less than 50%), and we want to compute Alice's odds for success or failure in this game.

What we are going to describe is a basic technique relevant not just to gambler's ruin, but to many much more complicated (and very realistic) games too. If a casino wants to know what odds to put on a given game, *it must do an analysis very much like what follows* (except that the numerical figures of \$2 and \$3 considered here would be rather different).

Remark 16.2.1. The mathematics of the situation we are considering is identical to that which arises in an entirely different context: understanding the effects of mutations over many generations via the Wright–Fisher model for genetic drift. Roughly speaking, here is the dictionary relating the gambling setup with the population dynamics setup: the total amount of money in the gambling setting is replaced with a fixed population size, flipping a coin is replaced with the passage of a specified genetic trait from each generation to the next, the probabilities 1/3 for each of heads and 2/3 for tails are replaced with probabilities p and $1 - p$ that an offspring inherits a certain genetic trait or not, and Alice's odds of eventually winning are replaced with the proportion of the population *in the long run* that has the genetic trait of interest.

We won't say more about the biological application here, but up to appropriate changes in terminology it is analyzed mathematically by exactly the same procedure that is used below. Of course, in the biological context one can seek a more robust framework that analyzes *several* genetic traits at once (this has no analogue in the win-lose gambling context). The mathematical model of Wright-Fisher can be generalized to accommodate this, and it is the focus of a lot of research in biology.

To model the game, we shall keep track of the *probability*, or *chance*, that Alice has a given amount of money. We turn it into a vector, keeping in mind that the total number of dollars in play is always $\$2 + \$3 = \$5$, so Alice cannot ever have more than \$5:

$$\mathbf{p} = \begin{bmatrix} \text{probability that Alice has \$0} \\ \text{probability that Alice has \$1} \\ \text{probability that Alice has \$2} \\ \text{probability that Alice has \$3} \\ \text{probability that Alice has \$4} \\ \text{probability that Alice has \$5} \end{bmatrix} \in \mathbf{R}^6.$$

Since it matters how many turns have been played, $\mathbf{p}$ is really a function $\mathbf{p}(n)$ of the number n of turns. Thus, for example, what we have said about the odds for the outcomes after the first turn amounts to saying

$$\mathbf{p}(1) = \begin{bmatrix} 0 \\ 1/3 \\ 0 \\ 2/3 \\ 0 \\ 0 \end{bmatrix}, \tag{16.2.1}$$

and it is convenient to make the definition $\mathbf{p}(0) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ since Alice *definitely* has \$2 when the game starts (the "zero" turn).

After n turns we have

$$\mathbf{p}(n) = \begin{bmatrix} p_0(n) \\ p_1(n) \\ p_2(n) \\ p_3(n) \\ p_4(n) \\ p_5(n) \end{bmatrix}$$

for some probabilities $p_i(n)$ that we do not (yet) know. (It is understood that if Alice ever actually reaches \$0 or \$5 then she remains in that state for the rest of time: the game ends, but to make the calculations go smoothly we assign such situations a 100% chance of remaining in that state for all future turns.) What happens in the next turn?

For example, how likely is Alice to have \$3 at the end of the $(n+1)$th turn, with $n \geq 1$? There are two ways this could happen:

- Alice had \$2 right after her nth turn, and *then* the coin came up tails. Chance: $p_2(n) \times \frac{2}{3}$.
- Alice had \$4 right after her nth turn, and *then* the coin came up heads. Chance: $p_4(n) \times \frac{1}{3}$.

This also holds for $n = 0$, by inspection of (16.2.1); think it through! In summary, the chance that Alice has \$3 right after her $(n+1)$th turn is $2p_2(n)/3 + p_4(n)/3$. In symbols: $p_3(n+1) = 2p_2(n)/3 + p_4(n)/3$ for all $n \geq 0$.

Likewise, for Alice to have \$0 right after the $(n+1)$th turn, with $n \geq 1$, the two possibilities leading to this are:

- Alice had \$0 right after her nth turn, and so stays in that "gambler's ruin" state forever. Chance: $p_0(n) \times 1 = p_0(n)$.
- Alice had \$1 right after her nth turn, and *then* the coin came up heads. Chance: $p_1(n) \times \frac{1}{3}$.

This also holds for $n = 0$, again by inspection of (16.2.1); convince yourself of this. Thus, $p_0(n+1) = p_0(n) + p_1(n)/3$ for all $n \geq 0$.

To similarly compute $p_1(n+1)$, we note that for Alice to have \$1 after the $(n+1)$th turn she must have had \$2 after the nth turn (and then then coin came up heads) since the alternative option of \$0 after the nth turn couldn't have happened (it would yield \$0 for all future terms). Hence, $p_1(n+1) = p_2(n) \times \frac{1}{3}$.

If you go through such reasoning for the other possible outcomes right after a given turn, and express the answer in matrix notation, you will find (please check for yourself):

$$\mathbf{p}(n+1) = \underbrace{\begin{bmatrix} 1 & 1/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 0 & 0 \\ 0 & 2/3 & 0 & 1/3 & 0 & 0 \\ 0 & 0 & 2/3 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 2/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2/3 & 1 \end{bmatrix}}_{M} \mathbf{p}(n) = M\,\mathbf{p}(n)$$

for all $n \geq 0$, with M denoting the indicated 6×6 matrix (e.g., the 4th row expresses the formula above for $p_3(n+1)$). The entries in each column of M are non-negative and sum to 1: another Markov matrix! A visual illustration of this matrix of probabilities is given in Figure 16.2.1.

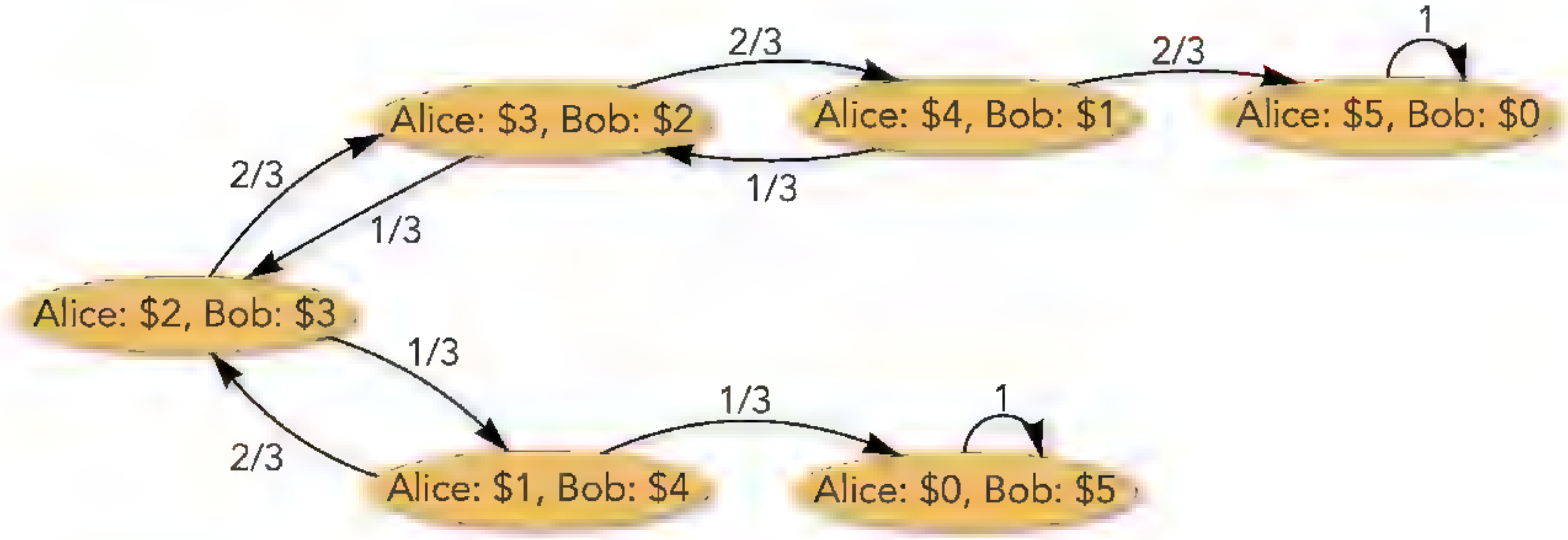


FIGURE 16.2.1. Odd for various outcomes of coin flips

Feeding the relation "$\mathbf{p}(n+1) = M\mathbf{p}(n)$" into itself repeatedly as in Section 16.1, we similarly obtain

$$\mathbf{p}(n) = M^n \mathbf{p}(0)$$

for all $n \geq 0$. In contrast with the bird migration example, for which high powers of the Markov matrix converge toward a matrix with all columns equal (as discussed in Remark 16.1.2), in this example the high powers M^n *do* converge toward a limiting matrix but without all columns equal:

$$M^n \to \begin{bmatrix} 1 & 0.4839 & 0.2258 & 0.0968 & 0.0323 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5161 & 0.7742 & 0.9032 & 0.9677 & 1 \end{bmatrix} \quad \text{as } n \to \infty$$

(the matrix entries are approximations to 4 decimal digits). The powers M^n become close to this limit already for $n = 30$. This stability for all large n is explained mathematically in Section D.2.

In concrete terms, the 0's in the intermediate rows of the limiting matrix express the extremely tiny odds of the game lasting a very long time (i.e., staying away from the terminating values \$0 and \$5), and the numbers along the top and bottom rows are the odds of Alice winding up with \$0 (top row) or \$5 (bottom row) after a long time: the entries are the odds of losing (top row) or winning (bottom row) when beginning at $\$j$ for $j = 0, 1, 2, 3, 4, 5$ (from left to right). More concretely,

$$\mathbf{p}(100) = M^{100}\mathbf{p}(0) \approx \begin{bmatrix} 1 & 0.4839 & 0.2258 & 0.0968 & 0.0323 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5161 & 0.7742 & 0.9032 & 0.9677 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \approx \begin{bmatrix} 0.2258 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.7742 \end{bmatrix}. \tag{16.2.2}$$

This calculations means that after very many turns, there is a chance of around 22.58% that Alice has \$0 (i.e., loses) and a chance of around 77.42% that Alice has \$5 (i.e., wins). Thus:

with around 22.6% probability, Alice loses; with around 77.4% probability, Alice wins.

For applications of these ideas to molecular structure, see Remark J.2.1. Probability theory gives alternative ways to figure out the preceding winning odds; see [KT, Sec. 3.6], especially [KT, (3.52)] with $N = 5$, $k = 3$.

Example 16.2.2. Suppose we want to compute the probability that the game ends before the 10th turn. How would we do this? Recall that if at any stage Alice reaches \$0 or \$5 then she remains in that state forever. Thus, the game ends before the 10th turn precisely when Alice is in the state of \$0 or \$5 after the 9th turn. (She may have reached that losing or winning state on an earlier turn, but then remains in it forever and so in particular is in that state after the 9th turn as well.)

Hence, this probability is $p_0(9) + p_5(9)$. In other words, we compute

$$\mathbf{p}(9) = M^9 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \approx \begin{bmatrix} 0.204 \\ 0.028 \\ 0 \\ 0.089 \\ 0 \\ 0.679 \end{bmatrix}, \quad \text{so } p_0(9) + p_5(9) \approx 0.204 + 0.679 = 0.883$$

(where again we do the computation on a computer). So the chance is approximately 88.3%. ■

Remark 16.2.3. In many real-world problems involving decisions made with uncertainty, the technique of Markov processes as above needs to be generalized to "Markov decision processes". This arises in many contexts, such as: finance (which and how much among a collection of stocks to purchase at a given time?), machine learning (how does a rover on Mars decide what to do given the available battery power and sunlight?), public health (how to determine quarantine levels at each stage of an epidemic?), and so on. See [Pu, Ch. 1] for a detailed discussion of many real-world examples.

Remark 16.2.4. The subject of "network analysis" (see Section 16.4 for a basic example) uses matrix powers, Markov matrices, and more sophisticated linear algebra to analyze information: the spread of disease through a population via probabilistic infection of one's contacts [BSH], managing transmission loads across networks (for power, telecommunications, etc.; see [BG, 3.3, 3.4, App. A to Ch. 3]), Netflix recommendations (based on what films have been viewed by customers who watch a specific film: see Example 21.6.3), and much more. In fact, the "Magna Carta of the Information Age" [Shan] (which provided the conceptual framework for all modern telecommunications and electronic manipulation of data) ultimately rests on treating communication of information in terms of a Markov chain (see [Shan, Secs. 4-6]).

16.3. Linear recurrence relations.

Next we consider more mathematical situation, involving a sequence of numbers in which each term is determined by the previous few terms according to a *linear* formula. This arises when modeling many real-world situations for which there is a feedback-loop mechanism, such as: population dynamics (a toy version of which was given in Section 16.1), search algorithms in computer science, digital signal processing, financial forecasting (via difference equations), and much more.

Consider the sequence of numbers starting with 1, 1, 1 for which each successive term is built from the previous three terms via the formula

$$a_n = a_{n-1} + 2a_{n-2} - a_{n-3} \text{ for } n \geq 4. \tag{16.3.1}$$

This sequence begins $1, 1, 1, 2, 3, 6, 10, 19, \ldots$; i.e., with $a_1 = 1, a_2 = 1, a_3 = 1, a_4 = 2, a_5 = 3, a_6 = 6$, and so on. The formula (16.3.1) expresses each term as a *linear function* of the previous 3 terms (this is called a "linear recurrence"). There is nothing special about using $1, 1, 1$ as the first three terms; we could have used any three numbers, but chose $1, 1, 1$ just to make the following discussion more concrete. In real life, the aspect that may become more sophisticated is to replace the specific linear recurrence (16.3.1) with one that involves more complicated coefficients on the right side.

Problem: Accurately approximate a_{1000}. (This is a prototype for the very natural task of qualitatively understanding long-term behavior in population dynamics, macroeconomics, etc.)

This problem doesn't seem to involve linear algebra, but understanding large powers of a fixed matrix turns out to be the key issue. To explain this, consider for each $n \geq 1$ the vector $\mathbf{v}(n) = \begin{bmatrix} a_n \\ a_{n+1} \\ a_{n+2} \end{bmatrix}$. Therefore,

$$\mathbf{v}(1) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{v}(2) = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}, \quad \mathbf{v}(3) = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \ldots.$$

The linear recurrence relation $a_{n+3} = a_{n+2} + 2a_{n+1} - a_n$ (i.e., (16.3.1) with $n+3$ in place of n) may be applied to the last entry in $\mathbf{v}(n+1)$ to obtain

$$\mathbf{v}(n+1) = \begin{bmatrix} a_{n+1} \\ a_{n+2} \\ a_{n+3} \end{bmatrix} = \begin{bmatrix} a_{n+1} \\ a_{n+2} \\ a_{n+2} + 2a_{n+1} - a_n \end{bmatrix}.$$

We can write this as a linear function of $\mathbf{v}(n)$:

$$\mathbf{v}(n+1) = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 2 & 1 \end{bmatrix}}_{A} \begin{bmatrix} a_n \\ a_{n+1} \\ a_{n+2} \end{bmatrix} = A\,\mathbf{v}(n);$$

this A is *not* a Markov matrix. Now feeding "$\mathbf{v}(n+1) = A\,\mathbf{v}(n)$" into itself repeatedly for decreasing values of n, we can make calculations such as this:

$$\mathbf{v}(10) = A\,\mathbf{v}(9) = A(A\,\mathbf{v}(8)) = A^2\,\mathbf{v}(8) = A^2(A\,\mathbf{v}(7)) = A^3\,\mathbf{v}(7) = \cdots = A^9\,\mathbf{v}(1) = A^9 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

More generally, such a calculation shows that

$$\begin{bmatrix} a_n \\ a_{n+1} \\ a_{n+2} \end{bmatrix} = \mathbf{v}(n) = A^{n-1} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

for any $n \geq 1$. This allows us to evaluate a_n as long as we know how to raise the matrix $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 2 & 1 \end{bmatrix}$ to powers. Given how complicated matrix multiplication appears to be, it may seem that to compute A^n for $n \approx 10$ is quite unpleasant by hand and for $n \approx 1000$ is a real nightmare. But remarkably, it turns out that there are elegant quick ways to accurately approximate A^n for large n (and even computers do *not* grind out matrix multiplication directly to compute large powers). This rests on notions to be introduced in Chapters 23 and 24, so we will return to this topic in Section 24.4.

Remark 16.3.1. The involvement of a big matrix power suggests that the a_n's grow exponentially in n. The situation is subtle because Sections 16.1 and 16.2 give interesting matrices whose big powers have all entries *not* growing at all. In Section 24.4 we will use linear algebra to understand the growth of big matrix powers and in Example 24.4.3 we will apply those methods to the powers of the 3×3 matrix A in the preceding discussion to obtain an exact formula for a_n yielding the approximation $a_n \approx c\lambda^n$ where $\lambda \approx 1.8109$ is the largest root of the polynomial $x^3 - x^2 - 2x + 1$ (the other two roots are approximately -1.2469 and $.4450$) and $c = -(5/7)\lambda^2 + (3/7)\lambda + 12/7 \approx 0.1672$. If you're curious, $a_{1000} \approx 9.1877 \times 10^{254}$ and more exactly it is

$$9187718511772266978178825494668544362076334204210134095096522796754406551731879708$$
$$4539126041948677028290950061074023021006381626356622940444747108801111817266390395$$
$$8925851861215584837681734828023270877310457406247208110156109651441897659179775342$$
$$361578394.$$

16.4. Friendship network. There are 7000 undergraduate students at Stanford. Let's arrange them in some order, from 1 to 7000. Let N be the 7000×7000 matrix whose ij-entry n_{ij} is 1 if student i and student j are friends and 0 if they are not. (Here we make the convention that a person cannot be a friend of themself, so all the diagonal entries of N are zero.)

Note that N is symmetric around its main diagonal (in the sense of Definition 15.1.1): the ij-entry and ji-entry are the same (both 0, or both 1) for any i and j. In real life the notion of being a friend may sometimes not be symmetric, but we ignore that. If instead one were to study the matrix N' encoding the relationship of being a follower (on some specific platform) rather than a friend, say N'_{ij} is 1 when student i follows student j and 0 otherwise, then symmetry fails for N'.

What information does N^2 (i.e., the matrix product $N\,N$) contain? That is, what is the meaning of the ij-entry of N^2? Using the formula (14.3.1), we find

$$ij\text{-entry of } N^2 = \sum_{k=1}^{7000} n_{ik} n_{kj}. \tag{16.4.1}$$

Now for each k, and each i and j, the product $n_{ik} \cdot n_{kj}$ is equal to either $1 \cdot 1 = 1$ or else $1 \cdot 0 = 0$ or $0 \cdot 1 = 0$ or $0 \cdot 0 = 0$, so it equals 1 precisely when $n_{ik} = 1 = n_{kj}$, which is to say student k is a friend of *both* student i and student j (in particular, this happens only when $k \neq i, j$; recall that the diagonal entries of N vanish by design). *Make sure you work through this and understand it!*

Thus, the kth term in the sum on the right side of (16.4.1) is equal to 0 or 1, and that term equals 1 precisely when student k is a friend with student i and with student j. The sum of those 1's is therefore the number of such students who are friends with student i and with student j, so

$$ij\text{-entry of } N^2 = \text{number of common friends of student } i \text{ and student } j.$$

In the special case $i = j$, this becomes: ii-entry of N^2 = number of friends of student i.

Example 16.4.1. Consider a toy version in which we select 5 Stanford undergraduates (labeled as students #1, #2, #3, #4, #5 in some way) and only study the friendship network among them. In this case we have a 5×5 matrix, say we call it N_{toy}, that could be something like this:

$$N_{\text{toy}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

This matrix encodes that student $\#2$ is friends with students $\#1$, $\#3$, and $\#4$ (so student 2 has 3 friends), and student $\#4$ is only friends with student $\#2$. This matrix is symmetric around its diagonal, just as we noted above for N (i.e., by flipping across the diagonal we recover N again, which just says visually that the ij-entry is the same as the ji-entry for all $i \neq j$, as is seen directly from how the matrix entries are defined: student i being a friend of student j is a "symmetric" relationship between i and j).

One computes

$$N_{\text{toy}}^2 = \begin{bmatrix} 2 & 0 & 2 & 1 & 0 \\ 0 & 3 & 0 & 0 & 2 \\ 2 & 0 & 2 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 2 & 0 & 0 & 2 \end{bmatrix},$$

so for example the number of friends of student $\#2$ is equal to the (2,2)-entry of N_{toy}^2 that is 3 (as we knew), the number of common friends of students $\#1$ and $\#3$ is the (1,3)-entry of N_{toy}^2 that is 2 (namely: students $\#2$ and $\#5$), and the number of common friends of students $\#1$ and $\#4$ is the (1,4)-entry of N_{toy}^2 that is 1 (namely: just student $\#2$). Note that N_{toy}^2 is also symmetric; this can be understood from the interpretation of the meaning of its ij-entry in terms of chains of friends linking student i to student j, but it is also a special case of a general fact concerning powers of *any* square matrix symmetric around its diagonal (see Example 20.3.6). ■

What about N^3; i.e, the matrix product $N\,N\,N$? By similar reasoning that we won't go through here, one finds that

$$ij\text{-entry of } N^3 = \text{number of } \textit{chains of friends of length } 3 \text{ between student } i \text{ and student } j;$$

i.e., student j is a friend of k who is a friend of ℓ who is a friend of i (this is called "three degrees of separation" since it says that j is a friend of a friend of a friend of student i). Since nobody is their own friend, in the chain "j is a friend of k who is a friend of ℓ who is a friend of i" necessarily $j \neq k$, $k \neq \ell$, and $\ell \neq i$, so if $i = j$ then all 3 values $i = j$, k, and ℓ are *different*. In contrast, when when $i \neq j$ then either all 4 values j, k, ℓ, i are different or else $j = \ell$ and $k - i$ (in which case there are exactly 2 different people in the chain, so then necessarily the different students i and j are friends since $i = k$ and $j = \ell$ with k and ℓ friends).

Example 16.4.2. How can we determine if every pair of students at Stanford is *at most* three degrees of separation away from each other? One might try to check if the entries of N^3 are all positive. But this is the wrong thing to do: the entries of N^3 count chains of friends of length *exactly* 3 and not *at most* 3. We need a way to count length-3 chains in which it is permitted for some student to be repeated as a friend of themself within the chain (e.g., turn a length-2 chain of friends in the initial setup where nobody is a friend of themself into a length-3 chain by repeating someone in the chain as their own friend).

This amounts to removing the original constraint of nobody being a friend of themself and instead *requiring* everyone to be their own friend: we make the diagonal entries all equal to 1 rather than all equal to 0. In other words, we work with the matrix $N + \mathrm{I}_{7000}$ rather than N, so we seek to determine if $(N + \mathrm{I}_{7000})^3$ has *all* entries > 0. (Note that for $i \neq j$, the ij-entry of $(N + \mathrm{I}_{7000})^3$ is *not* the number of different ways of getting from student i to student j through friendship chains of at most 3. Indeed, some shorter chains become length-3 chains in many ways. For example, if i and j are already friends then the 1-step chain from i to j underlies *many* 3-step chains, such as: from i to i to i to j, and from i to i to j to j, and even from i to j to i to j.)

In our toy situation with 5 students corresponding to N_{toy} in Example 16.4.1, we have

$$(N_{\mathrm{toy}} + \mathrm{I}_5)^3 = \begin{bmatrix} 7 & 8 & 6 & 3 & 7 \\ 8 & 10 & 8 & 6 & 6 \\ 6 & 8 & 7 & 3 & 7 \\ 3 & 6 & 3 & 4 & 2 \\ 7 & 6 & 7 & 2 & 7 \end{bmatrix},$$

so we conclude from the positivity of all entries that indeed there are at most "3 degrees of separation" among these 5 students. In contrast, if you calculate $(N_{\mathrm{toy}} + \mathrm{I}_5)^2$ then you'll find that the $(4, 5)$-entry (and also the $(5, 4)$-entry) is equal to 0, so 2 degrees of separation do not suffice for this population of 5 students (we cannot get from student #4 to student #5 in at most 2 steps through a chain of friends). ■

Chapter 16 highlights (links to highlights in [illegible] and next chapters)

Notation	Meaning	Location in text
nothing new!		

Concept	Meaning	Location in text
Markov matrix	a square matrix whose entries are non-negative and columns each sum to 1	Section 16.1

Result	Meaning	Location in text
some pop. dynamics are governed by matrix powers	for a collection of populations moving around with fixed "transition rates" at each time step, long-term behavior is governed by powers of an associated matrix	Section 16.1
high powers of certain Markov matrices have all columns nearly the same	if a Markov matrix M has all entries positive (or some equal to 0 subject to additional hypotheses) then for big r the columns of M^r are all nearly the same	Remark 16.1.2
some gambling outcomes are governed by matrix powers	odds of winning by a certain stage or in the long run in certain games of chance are governed by powers of a matrix of "transition probabilities"	Section 16.2

Skill	Location in text
for certain problems about population dynamics or gambling, set up relevant matrix language to analyze with a matrix power and know that high powers encode long-term behavior (not expected to compute high powers yourself)	Sections 16.1, 16.2

17. Multivariable Chain Rule

Our next application of matrix multiplication is to adapt "$dy/dx = (dy/du)(du/dx)$" from single-variable functions to multivariable functions by using matrix multiplication.

By the end of this chapter, you should be able to:

(CR1) translate a word problem involving composite functions into a mathematical statement;
(CR2) use the multivariable Chain Rule to compute the derivative matrix of a composite function as a matrix product, with careful attention to the *order of multiplication* of matrices;
(CR3) recognize the utility of the multivariable Chain Rule for computing the derivative of a single-variable function $\mathbf{R} \to \mathbf{R}$ that is expressed as a composition $\mathbf{R} \to \mathbf{R}^n \to \mathbf{R}$.

Instances of (CR2) arise *everywhere* in physics and engineering when expressing differential equations in new coordinate systems (e.g., passing from rectangular to spherical or cylindrical coordinates). Let's comment on (CR3). A function $f : \mathbf{R} \to \mathbf{R}$ of time may depend on many inputs that also depend on time; this involves expressing f as a composition $\mathbf{R} \to \mathbf{R}^n \to \mathbf{R}$. The multivariable Chain Rule then expresses df/dt in terms of the rates of change (in time) of the inputs. Examples include:

(i) total profit as a function of production costs, total sales, and selling price,
(ii) population size as a function of many biological and sociological factors,
(iii) the backpropagation algorithm that underlies "learning from errors" in artificial neural networks to compute "weights" that power chatbots (referred to as the "crown jewels" for such AI systems in this article); see Appendix G for more details on this.

17.1. Statement of main formula.

If f and g are functions $\mathbf{R} \to \mathbf{R}$, their *composition* $f \circ g : \mathbf{R} \to \mathbf{R}$ is defined by the rule $(f \circ g)(x) = f(g(x))$. Many familiar functions are built as compositions of simpler ones. (The function $f \circ g$ is often referred to as a *composite* function.) In Section 8.2 we gave a few examples and some illustrations of the ubiquitous role of the composition operation when forming new functions from simpler ones.

Theorem 17.1.1 (Single-Variable Chain Rule). The derivative $(f \circ g)'(x) = (f(g(x)))'$ of the composition $f \circ g$ is equal to the product function $f'(g(x))\, g'(x)$. Equivalently, the value $(f \circ g)'(a)$ of the derivative of $f \circ g$ at $x = a$ is the product $f'(g(a))\, g'(a)$ for every $a \in \mathbf{R}$.

This is often written in more suggestive terms as

$$\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx}$$

with the shorthand that y stands for f and u stands for g, engaging in some abuse of notation (that you got accustomed to through practice): "y" on the left side is really $y \circ u$ as a function of x, and "u" in the first factor on the right side is an independent variable whereas in the second factor on the right side it is not an independent variable but rather is a function of x!

The experience in single-variable calculus that derivatives are *much easier* to compute than anti-derivatives ultimately stems from the fact that such a simple universal formula expresses the derivative of a composite function in terms of the derivatives of the functions being composed (whereas there is nothing like this available for anti-derivatives). One thinks of the Chain Rule as a recipe for computing the derivative of $f \circ g$ at any point $x = a$ in terms of two ingredients:

- the derivative f' evaluated at the point $g(a)$,
- the derivative g' evaluated at the point a.

Once these ingredients are worked out, the derivative $(f \circ g)'$ at $x = a$ is obtained by multiplying them together. With some practice, in single-variable calculus you learned how to apply the Chain Rule to compute a huge range of derivatives. Realistic mathematical models often involve interdependencies of input expressed via functions of *several* variables, such as listed near the start of this chapter.

Example 17.1.2. Suppose a person walks on a mountain with height function $h(x, y) \in \mathbf{R}$ above sea level where x and y are the (horizontal) distances in feet respectively east and north of the summit (e.g., $x < 0$ refers to being west of the summit); the mountain is the graph of h in the sense of Section 9.5. As a function of time t in minutes, let $p(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \in \mathbf{R}^2$ be the path traversed by the person (i.e., at time t their east-west displacement from the summit is $x(t)$ and their north-south displacement from the summit is $y(t)$). Thus, $a(t) = h(p(t))$ denotes the person's altitude above sea level at time t.

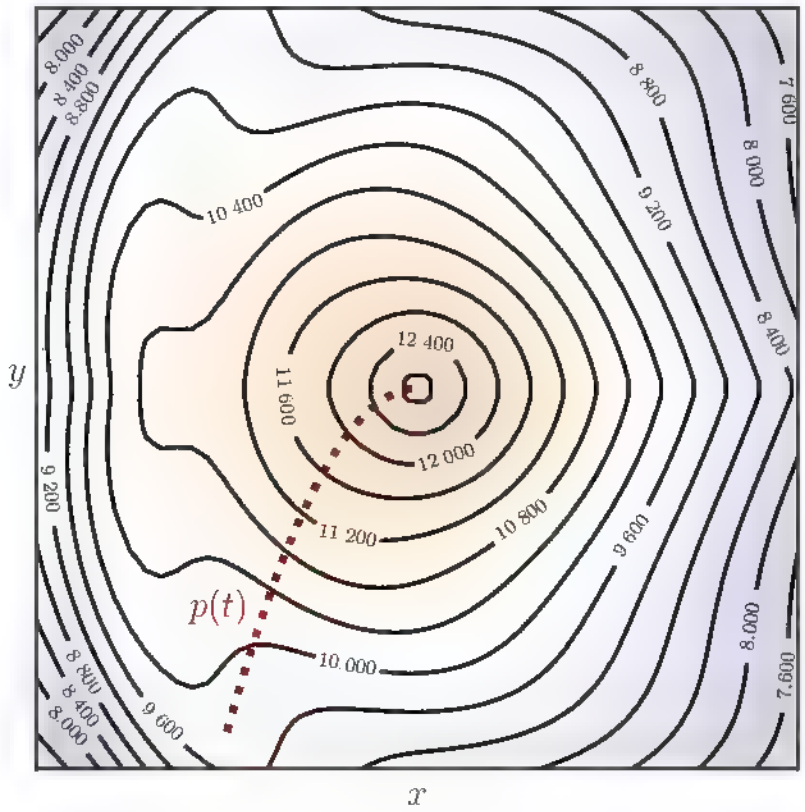


FIGURE 17.1.1. Contour plot showing a red path to the top of mountain

We wish to compute $a'(t)$, the rate of change of altitude as a function of time. Since a is a function $\mathbf{R} \to \mathbf{R}$, the task of computing $a'(t)$ is something we could imagine trying to do in the context of single-variable calculus. But there is *nothing* in single-variable calculus telling us how $a'(t)$ can be *completely determined* by two rather different ingredients: the hilliness of the mountain and the xy-velocities of motion.

The derivative matrix $(Dh)(x, y) = \begin{bmatrix} \partial h/\partial x & \partial h/\partial y \end{bmatrix}$ keeps track of how hilly the mountain is. For instance, $\partial h/\partial y$ measures steepness of the mountain in the north-south direction (e.g., if very negative then it drops quickly for someone walking in a northerly direction). The derivative matrix $(Dp)(t) = \begin{bmatrix} x'(t) \\ y'(t) \end{bmatrix}$ keeps track of the speed in east-west and north-south directions (e.g., if $x'(t) < 0$ then the person is walking in a westerly direction); one might also regard $(Dp)(t)$ as a "velocity vector" for the xy-directions. The multivariable Chain Rule to be stated and illustrated below will give a *universal formula* (irrespective of the specifics of h and p) for $a'(t)$ in terms of $(Dh)(x, y)$ and $(Dp)(t)$. ■

Example 17.1.3. In the context of Lagrange multipliers from Chapter 12, we noted in Remark 12.2.14 that when maximizing a function $f(\mathbf{x})$ subject to a constraint $g(\mathbf{x}) = c$, one may regard the point $\mathbf{a}$ at which

f attains its maximum subject to the constraint $g = c$ as a vector-valued function of c. This maximum is $f(\mathbf{a}(c))$, which we now see is a *composition* of two functions: first we have $\mathbf{a}$ as a function of c, and then we plug that into f. Hence, the ordinary calculus derivative $\frac{d}{dc}(f(\mathbf{a}(c)))$ expressing the rate of change in c for the maximum of $f(\mathbf{x})$ subject to $g(\mathbf{x}) = c$ is a situation similar in form to the one in Example 17.1.2.

It was mentioned in Remark 12.2.14 that the Lagrange multiplier $\lambda(c)$ associated with this constrained optimization problem is exactly that derivative $\frac{d}{dc}(f(\mathbf{a}(c)))$. To explain why this works, we need a way to compute such composite derivatives which involve a *vector-valued* intermediate step (namely, $\mathbf{a}$). We will carry this out in Example 17.1.9. ■

Example 17.1.4. The output of a mechanical or biological or financial process often rests upon many ingredients, each of which depend on many other ingredients, and so on, setting up a chain of many layers of dependence. For example, the velocity of a runner depends on energy stored in various muscles and signals sent from neurons, and these in turn depend on air intake, oxygen levels in blood, the hilliness of the path being traversed, etc. It is important to understand how the output varies in terms of such deeper ingredients, which is exactly working with a *composition* of multivariable functions.

In general, we have a vector-valued function $\mathbf{F}(x_1, \dots, x_m)$ of several inputs x_j each of which depends on a collection of other inputs $(y_1, \dots, y_p)$, and *those in turn* may depend on yet other inputs $(z_1, \dots, z_n)$, and so on. When the first step of dependencies is plugged in so $\mathbf{F}$ is expressed in terms of the y_k's, or we go a step further and express $\mathbf{F}$ in terms of the z_r's (or beyond), we obtain a multivariable *composition*. Can we express the partial derivatives of each component function of the composition in terms of some "universal formula" built out of the partial derivatives of the dependence functions at each layer (e.g., x_j's in terms of y_k's, y_k's in terms of z_r's) *without explicitly plugging all layers of dependence into* $\mathbf{F}$?

Here is an example. Consider $\mathbf{F}(v, w) = (vw, v + w)$ with a chain of dependencies on other parameters

$$(v, w) = (xy, 3x + yz), \quad (x, y, z) = (st, s^2t^2, s + t),$$

and suppose we want to analyze the behavior of $\mathbf{F}$ when expressed in terms of (x, y, z) or (s, t). We can express such dependencies in terms of the functions

$$\mathbf{R}^2 \xrightarrow{\mathbf{h}} \mathbf{R}^3 \xrightarrow{\mathbf{g}} \mathbf{R}^2 \xrightarrow{\mathbf{f}} \mathbf{R}^2$$

defined by

$$\mathbf{h}(s, t) = (st, s^2t^2, s + t), \quad \mathbf{g}(x, y, z) = (xy, 3x + yz), \quad \mathbf{f}(v, w) = (vw, v + w),$$

namely "$\mathbf{F}$ in terms of (x, y, z)" is $\mathbf{f} \circ \mathbf{g}$, and "$\mathbf{F}$ in terms of (s, t)" is $\mathbf{f} \circ \mathbf{g} \circ \mathbf{h}$. (Make sure you understand this!) A typical question we would like to systematically be able to answer is this: when $\mathbf{F}$ is viewed as a function of (s, t), what is $(\partial F_1/\partial t)\big|_{(2,\ 1)}$ (where F_1 is the first component function of $\mathbf{F}$)?

One way to do this is raw substitution: we plug in explicitly to get

$$\begin{aligned} \mathbf{F}(s, t) = \mathbf{f}(\mathbf{g}(\mathbf{h}(s, t))) &= \mathbf{f}(\mathbf{g}(st, s^2t^2, s + t)) \\ &= \mathbf{f}(s^3t^3, 3st + s^3t^2 + s^2t^3) \\ &= (3s^4t^4 + s^6t^5 + s^5t^6, s^3t^3 + 3st + s^3t^2 + s^2t^3), \end{aligned}$$

so $F_1(s, t) = 3s^4t^4 + s^6t^5 + s^5t^6$ and hence $\partial F_1/\partial t = 12s^4t^3 + 5s^6t^4 + 6s^5t^5$. Evaluating at $(s, t) = (2, -1)$, we get $-12 \cdot 16 + 5 \cdot 64 - 6 \cdot 32 = -64$.

But this is a bit tedious, and for many practical situations it would be very unpleasant or impossible to proceed this way. For instance, in the context of a neural network for machine learning, such partial derivatives need to be computed but the number of variables at each layer could be on the order of *millions* and the number of layers (i.e., the number of composition steps) can also be rather large (far more than composing just three functions as above). After we introduce the multivariable Chain Rule, we'll see in

Example 17.1.6 how it works out for the example above. (In the optional Appendix G, for those who are interested, we discuss in more detail the tremendously important application to neural networks.) ■

Theorem 17.1.5 (The multivariable Chain Rule). If $\mathbf{f} : \mathbf{R}^p \to \mathbf{R}^m$ and $\mathbf{g} : \mathbf{R}^n \to \mathbf{R}^p$ are two functions then the derivative matrix of $\mathbf{f} \circ \mathbf{g} : \mathbf{R}^n \to \mathbf{R}^m$ at a point $\mathbf{a} \in \mathbf{R}^n$ is computed in terms of the derivative matrices of $\mathbf{f}$ and $\mathbf{g}$ at appropriate points via the formula

$$(D(\mathbf{f} \circ \mathbf{g}))(\mathbf{a}) = (D\mathbf{f})(\mathbf{g}(\mathbf{a}))\,(D\mathbf{g})(\mathbf{a})$$

where the right side is a **matrix** product. Working at a variable point $\mathbf{v} = (v_1, \dots, v_n) \in \mathbf{R}^n$, this takes on the equivalent form

$$(D(\mathbf{f} \circ \mathbf{g}))(\mathbf{v}) = (D\mathbf{f})(\mathbf{g}(\mathbf{v}))\,(D\mathbf{g})(\mathbf{v}).$$

Warning: Most multivariable calculus books do not state this matrix version of the Chain Rule. They express the result in terms of the matrix entries (partial derivatives), which has the appearance of a different result for every possible number of variables. The uniform matrix version we have presented illustrates that "multivariable calculus is just like single-variable calculus if you use matrices." There is only one Chain Rule: the version we have written above. (For the formulation in terms of matrix entries, see (17.1.3) when $n = m = 1$ and Remark 17.1.10 (or (17.4.2)) for the general case.)

The intuition behind the multivariable Chain Rule is rather concrete, as follows. By definition, $(D\mathbf{f})(\mathbf{a})$ is the best linear approximation to $\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{a})$ for $\mathbf{x}$ near $\mathbf{a}$. And by definition **matrix multiplication calculates exactly the composition of linear transformations**. Thus, in words this Chain Rule says:

> *the best linear approximation to a composite function* $\mathbf{f} \circ \mathbf{g}$ *at* $\mathbf{a} \in \mathbf{R}^n$ *is the composition (in the same order!) of the best linear approximations to* $\mathbf{f}$ *(at* $\mathbf{g}(\mathbf{a})$*) and* $\mathbf{g}$ *(at* $\mathbf{a}$*).*

In Section 17.4 we explain more fully this linear algebra intuition for where the multivariable Chain Rule comes from, for those who wish to understand it in more detail. (We will later see a reformulation of this Chain Rule in terms of somewhat complicated formulas with scalar-valued component functions, in (17.1.4) and (17.1.6), but only the perspective of matrix multiplication provides a genuine explanation for where such scalar-valued formulas come from.)

For $\mathbf{v} \in \mathbf{R}^n$, all participants in the multivariable Chain Rule formula are matrices whose entries are functions of $v_1, \dots, v_n$ (the coordinates of the input $\mathbf{v} \in \mathbf{R}^n$). Note that the matrix multiplication in the formula "makes sense" in terms of the number of rows and columns in the respective matrices:

$$\underbrace{(D(\mathbf{f} \circ \mathbf{g}))(\mathbf{v})}_{m \times n} = \underbrace{(D\mathbf{f})(\mathbf{g}(\mathbf{v}))}_{m \times p}\ \underbrace{(D\mathbf{g})(\mathbf{v})}_{p \times n}$$

Example 17.1.6. Let's revisit Example 17.1.4 and compute $(\partial F_1/\partial t)(2, -1)$ via the Chain Rule. In terms of the notation introduced there, we seek to compute the $(1, 2)$-entry (i.e., upper-right entry) of the 2×2 matrix $(D(\mathbf{f} \circ \mathbf{g} \circ \mathbf{h}))(2, -1)$. This 2×2 derivative matrix is equal to

$$(D\mathbf{f})(\mathbf{g}(\mathbf{h}(2, -1)))\,(D\mathbf{g})(\mathbf{h}(2, -1))\,(D\mathbf{h})(2, -1) = (D\mathbf{f})(-8, -2)\,(D\mathbf{g})(-2, 4, 1)\,(D\mathbf{h})(2, -1).$$

Here, the point at which each derivative matrix is being evaluated is obtained by applying the chain of compositions to $(2, -1)$:

$$(2, -1) \overset{\mathbf{h}}{\rightsquigarrow} (-2, 4, 1) \overset{\mathbf{g}}{\rightsquigarrow} (-8, -2)$$

via the formulas defining $\mathbf{g}$ and $\mathbf{h}$. The formulas defining $\mathbf{f}$, $\mathbf{g}$, and $\mathbf{h}$ give

$$D\mathbf{f} = \begin{bmatrix} w & v \\ 1 & 1 \end{bmatrix}, \quad D\mathbf{g} = \begin{bmatrix} y & x & 0 \\ 3 & z & y \end{bmatrix}, \quad D\mathbf{h} = \begin{bmatrix} t & s \\ 2st^2 & 2s^2t \\ 1 & 1 \end{bmatrix}.$$

Evaluating these at the required points then gives that $(D\mathbf{F})(2,-1)$ is the matrix product

$$\begin{bmatrix} -2 & -8 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 4 & -2 & 0 \\ 3 & 1 & 4 \end{bmatrix}\begin{bmatrix} -1 & 2 \\ 4 & -8 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} -32 & -4 & -32 \\ 7 & -1 & 4 \end{bmatrix}\begin{bmatrix} -1 & 2 \\ 4 & -8 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} -16 & -64 \\ -7 & 26 \end{bmatrix}.$$

The desired $(1,2)$-entry is -64 (agreeing with what we obtained in Example 17.1.4). Observe that this method is *completely different* from the bare-hands approach in Example 17.1.4. ■

Example 17.1.7. In the setting of Example 17.1.2, let's see what we can learn from the multivariable Chain Rule. We expressed the altitude $a : \mathbf{R} \to \mathbf{R}$ above sea level at time t as the composition $\mathbf{R} \xrightarrow{p} \mathbf{R}^2 \xrightarrow{h} \mathbf{R}$. The multivariable Chain Rule says that at a time t_0, the 1×1 matrix $[a'(t_0)] = (Da)(t_0)$ equals the product

$$(Dh)(p(t_0))(Dp)(t_0) = \begin{bmatrix} h_x(p(t_0)) & h_y(p(t_0)) \end{bmatrix}\begin{bmatrix} x'(t_0) \\ y'(t_0) \end{bmatrix} = \begin{bmatrix} h_x(p(t_0))\,x'(t_0) + h_y(p(t_0))\,y'(t_0) \end{bmatrix}.$$

Comparing entries of the (equal) 1×1 matrices $(Da)(t_0)$ and $(Dh)(p(t_0))(Dp)(t_0)$ yields

$$a'(t_0) = h_x(p(t_0))\,x'(t_0) + h_y(p(t_0))\,y'(t_0) = (\nabla h)(p(t_0)) \cdot p'(t_0) \tag{17.1.1}$$

where $p'(t) = \begin{bmatrix} x'(t) \\ y'(t) \end{bmatrix}$ is the velocity at time t. For example, after 10 minutes the rate of change of the altitude of the person's path is $a'(10) = h_x(p(10))\,x'(10) + h_y(p(10))\,y'(10)$. To think more broadly in terms of general time t (not focusing on a specific time value), we rewrite (17.1.1) as

$$\frac{da}{dt} = \frac{\partial h}{\partial x}\frac{dx}{dt} + \frac{\partial h}{\partial y}\frac{dy}{dt} = (\nabla h)(p(t)) \cdot p'(t) \tag{17.1.2}$$

(where it is understood that the partial derivatives of h are really being evaluated at the point $p(t) = (x(t), y(t))$). This is a "universal formula" for how rapidly the person is moving up and down in altitude in terms of two ingredients: (i) their (signed) speeds in the x-direction and y-direction, (ii) the geometric information of the steepness of the mountain. Giving a "universal formula" such as (17.1.2) for a rate of change of a composition (such as $a(t) = (h \circ p)(t)$) in terms of rates of change for intermediate layers of a composition (such as for h and p) is the content of the multivariable Chain Rule.

Let's see what this is saying in a specific example. Suppose the mountain has an elliptical level curve at each elevation; for specificity, suppose the height $h(x,y)$ of the mountain over a point (x,y) at sea level (with the summit over $(0,0)$) is $h(x,y) = 200 - 3x^2 - xy - y^2$, so the height at the summit is $h(0,0) = 200$ and in general the part of the mountain at a height c is the level curve $200 - 3x^2 - xy - y^2 = c$ that is a tilted ellipse as shown in Figure 17.1.2 (which is a contour plot of h). For the path up the hill seen from above (i.e., as if drawn on a contour plot, as in Figure 17.1.2), suppose it is $p(t) = \begin{bmatrix} t - 20 \\ -3t + 60 \end{bmatrix}$ (so the actual position in space is at an altitude $a(t) = h(t-20, -3t+60) = h(p(t))$ above the point $(t-20, -3t+60)$ at sea level). From a bird's-eye view, the path of motion looks like a straight line with constant speed in the xy-directions, with initial position $p(0) = \begin{bmatrix} -20 \\ 60 \end{bmatrix}$ and placement at the summit at $t = 20$ minutes since $p(t) = (0,0)$ for $t = 20$. Let's work out what (17.1.2) is saying in this case.

We have

$$(Dh)(x,y) = \begin{bmatrix} -6x - y & -x - 2y \end{bmatrix}, \quad (Dp)(t) = \begin{bmatrix} 1 \\ -3 \end{bmatrix}$$

(so the velocity in the xy-directions is constant in time). Substituting this into (17.1.2) yields

$$a'(t) = (-6x - y)(1) + (-x - 2y)(-3) = -6x - y + 3x + 6y = -3x + 5y$$

where really $(x, y) = p(t)$, so we should substitute $x(t) = t - 20$ for x and $y(t) = -3t + 60$ for y to get

$$a'(t) = -3(t - 20) + 5(-3t + 60) = -18t + 360.$$

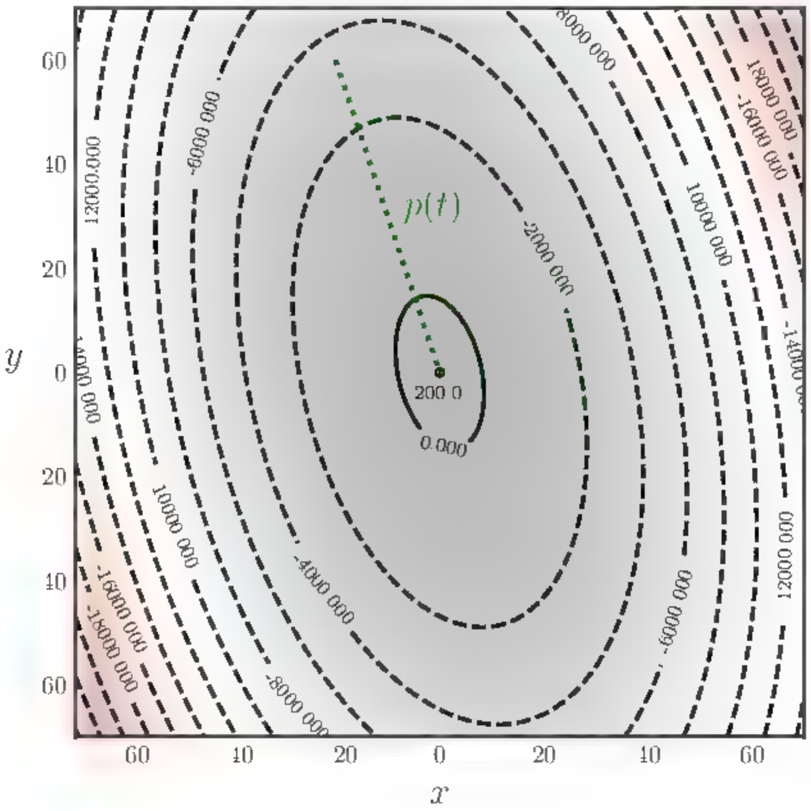


FIGURE 17.1.2. Green path directly up a mountain that is an ellipse at each elevation

As a safety check, let's confirm that this computation of $a'(t)$ using the "universal formula" (17.1.2) agrees with what we get by explicit substitution:

$$\begin{aligned}
a(t) = h(p(t)) &= h(t - 20, -3t + 60) \\
&= 200 - 3(t - 20)^2 - (t - 20)(-3t + 60) - (-3t + 60)^2 \\
&= 200 - 3(t^2 - 40t + 400) - (-3t^2 + 120t - 1200) - (9t^2 - 360t + 3600) \\
&= 200 - 3t^2 + 120t - 1200 + 3t^2 - 120t + 1200 - 9t^2 + 360t - 3600 \\
&= -9t^2 + 360t - 3400,
\end{aligned}$$

so $a'(t) = -18t + 360$ once again. Observe that there was some tedium involved in working out $a(t)$ explicitly, whereas our computation of $a'(t)$ via the multivariable Chain Rule in the guise of (17.1.2) *entirely bypassed* the need to explicitly compute $a(t)$, instead expressing $a'(t)$ in terms of the derivatives of h and p that are easier to work with. ■

The algebra involved in computing $a'(t)$ via the multivariable Chain Rule is ultimately less than the fully explicit second way of computing it. But the real significance of the Chain Rule goes beyond being a time-saving device, or a way to simplify calculations one does by hand: it provides a formula as in (17.1.2) that tells us in precise quantitative terms exactly how the rate of change of a composite procedure (such as $a = h \circ p$) is *completely determined* by knowledge of rates of change arising from the various inputs (such as $p(t)$ and $h(x, y)$).

That enables us to draw *qualitative conclusions* for how the overall rate of change is impacted when we make adjustments to the input. There are numerous situations in economics, the natural sciences, computer science, and so on where an overall effect is described in terms of many inputs, and in such cases the multivariable Chain Rule is the best tool for analyzing how the rate of change of an overall effect depends on that of the inputs.

For instance, suppose someone else moves twice as quickly in the x-direction and three times as quickly in the y-direction. Their path $P(t)$ is $\begin{bmatrix} 2x(t) \\ 3y(t) \end{bmatrix}$, so their altitude function $A(t) = h(P(t))$ has rate of change in time given by

$$A'(t) = \frac{\partial h}{\partial x}\frac{d(2x)}{dt} + \frac{\partial h}{\partial y}\frac{d(3y)}{dt} = 2\frac{\partial h}{\partial x}\frac{dx}{dt} + 3\frac{\partial h}{\partial y}\frac{dy}{dt}.$$

Comparing this with (17.1.2), we see that there isn't any "easy" direct formula for $A'(t)$ in terms of $a'(t)$ (because the speed-up factors of 2 and 3 in the x and y directions are not the same), but nonetheless we can see qualitatively how $A'(t)$ deviates from $a'(t)$ in terms of how the formulas for them deviate from each other. This qualitative comparison of $A'(t)$ and $a'(t)$ is the same *regardless* of the specific details of the path p or the hilliness h of the mountain.

At first glance, the general multivariable Chain Rule may seem like a complicated formula, but its formal structure is *pretty much the same* as the single-variable Chain Rule you already know! The two ingredients in the multivariable Chain Rule are:

- the derivative matrix $D\mathbf{f}$ evaluated at $\mathbf{g}(\mathbf{a})$ (or at $\mathbf{g}(\mathbf{v})$ if working at a variable point $\mathbf{v}$ in $\mathbf{R}^n$),
- the derivative matrix $D\mathbf{g}$ evaluated at $\mathbf{a}$ (or at $\mathbf{v}$ if working at a variable point in $\mathbf{R}^n$).

We emphasize:

the derivative matrix for $\mathbf{f} \circ \mathbf{g}$ at the point of interest is obtained by multiplying the derivative matrices of the two ingredients *in the same order* as the order of composition of the functions (i.e., $\mathbf{f}$-derivative on the left, $\mathbf{g}$-derivative on the right).

Matrix multiplication is sensitive to the order of multiplication except for 1×1 matrices (which is scalar multiplication by another name), and the single-variable Chain Rule is the *special case* of the multivariable Chain Rule with 1×1 matrices. For this reason, **from now on "Chain Rule" will mean the multivariable Chain Rule.**

Example 17.1.8. The formula in (17.1.2) for $n = m = 1$ with $p = 2$ generalizes to $n = m = 1$ and any p as follows. Suppose a function $\mathbf{R} \to \mathbf{R}$ is expressed as a composition $f \circ \mathbf{g}$ for some $\mathbf{g} : \mathbf{R} \to \mathbf{R}^p$ and $f : \mathbf{R}^p \to \mathbf{R}$. The case $p = 2$ arose in Example 17.1.2, and examples with larger p pervade the applications mentioned near the start of this chapter. (See Appendix G for striking applications with more general n and m.) Denote the component functions of $\mathbf{g}$ as $g_1, \dots, g_p$ (i.e., $\mathbf{g}(t) = (g_1(t), \dots, g_p(t))$), and consider f as a function of $\mathbf{y} = (y_1, \dots, y_p) \in \mathbf{R}^p$.

The deduction of (17.1.2) works equally well for any p, as follows. The unique entry $(f \circ \mathbf{g})'(t)$ in the 1×1 matrix $(D(f \circ \mathbf{g}))(t)$ can be computed via the Chain Rule from the matrix product

$$(Df)(\mathbf{g}(t))(D\mathbf{g})(t) = \begin{bmatrix} \frac{\partial f}{\partial y_1} & \frac{\partial f}{\partial y_2} & \cdots & \frac{\partial f}{\partial y_p} \end{bmatrix} \begin{bmatrix} g_1'(t) \\ g_2'(t) \\ \vdots \\ g_p'(t) \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial y_1}\frac{dg_1}{dt} + \cdots + \frac{\partial f}{\partial y_p}\frac{dg_p}{dt} \end{bmatrix}$$

where it is understood that each $\partial f/\partial y_k$ is evaluated at $\mathbf{g}(t)$. Comparing the entries in the 1×1 matrices then gives the equality of numbers

$$(f \circ \mathbf{g})'(t) = \sum_{k=1}^{p} \frac{\partial f}{\partial y_k}\frac{dg_k}{dt} = (\nabla f)(\mathbf{g}(t)) \cdot \mathbf{g}'(t) \tag{17.1.3}$$

where $\mathbf{g}'(t)$ is the "velocity vector" in $\mathbf{R}^p$ with kth entry $g_k'(t)$ and each $\partial f/\partial y_k$ is evaluated at $\mathbf{g}(t)$ (recovering (17.1.2) when $p = 2$, up to a change in the letters used to denote various functions, and recovering the Chain Rule in single-variable calculus when $p = 1$). Here is a notationally more convenient formulation in the spirit of (17.1.2):

$$\frac{df}{dt} = \sum_{k=1}^{p} \frac{\partial f}{\partial y_k}\frac{dy_k}{dt} \tag{17.1.4}$$

where $y_k(t)$ denotes the kth component function of $\mathbf{g}$ (i.e., $\mathbf{g}(t) = (y_1(t), \dots, y_p(t))$).

■

Example 17.1.9. As an application of (17.1.3) (or (17.1.4)), we finish off Example 17.1.3: in the notation there, we wanted to compute the derivative with respect to c of $f(\mathbf{a}(c))$ and show it equals $\lambda(c)$. Since we regard c as a varying quantity, we rename it as t.

The formula (17.1.3) tells us that the derivative of $f(\mathbf{a}(t))$ with respect to t is the dot product of the gradient $(\nabla f)(\mathbf{a}(t))$ at $\mathbf{a}(t)$ against the vector $\mathbf{a}'(t)$ whose jth entry is $a_j'(t)$. The main result on Lagrange multipliers (Theorem 12.2.1) tells us that $(\nabla f)(\mathbf{a}) = \lambda\,(\nabla g)(\mathbf{a})$ (ignoring the case $(\nabla g)(\mathbf{a}) = \mathbf{0}$). Remembering the dependence of everything on the choice of value of t, this really says

$$(\nabla f)(\mathbf{a}(t)) = \lambda(t)\,(\nabla g)(\mathbf{a}(t)).$$

Putting it all together, we have

$$\frac{d}{dt}(f(\mathbf{a}(t))) = (\nabla f)(\mathbf{a}(t)) \cdot \mathbf{a}'(t) = (\lambda(t)\,(\nabla g)(\mathbf{a}(t))) \cdot \mathbf{a}'(t) = \lambda(t)((\nabla g)(\mathbf{a}(t)) \cdot \mathbf{a}'(t)).$$

Ah, but the dot product $(\nabla g)(\mathbf{a}(t)) \cdot \mathbf{a}'(t)$ on the right side is the same type of multivariable Chain Rule output we have already encountered, except with g in the role of f. More specifically, this is the derivative with respect to t of the composite function $g(\mathbf{a}(t))$ (rather than $f(\mathbf{a}(t))$ as above). But the whole point of t is that it keeps track of the *constraint* on g! More specifically, by design $\mathbf{a}(t)$ is a point on the constraint locus $g(\mathbf{x}) = t$, so $g(\mathbf{a}(t)) = t$. So we conclude that

$$\frac{d}{dt}(f(\mathbf{a}(t))) = \lambda(t)\,\frac{d}{dt}(g(\mathbf{a}(t))) = \lambda(t)\,\frac{d}{dt}(t) = \lambda(t)\,(1) = \lambda(t),$$

as desired. ■

Remark 17.1.10. There is a generalization of (17.1.4) for the general "scalar-valued" case $f : \mathbf{R}^p \to \mathbf{R}$ that uses $\mathbf{g} : \mathbf{R}^n \to \mathbf{R}^p$ for any $n \geq 1$. It is a restatement of the multivariable Chain Rule in the case $m = 1$ avoiding the language of matrices; this is the version stated in many multivariable calculus textbooks, and it is widely used in scientific and engineering literature, so we now present it.

Consider $\mathbf{g} : \mathbf{R}^n \to \mathbf{R}^p$, and write its components as $g_1(\mathbf{x}), \dots, g_p(\mathbf{x})$. For any $f : \mathbf{R}^p \to \mathbf{R}$, the Chain Rule formula for $f \circ \mathbf{g}$ can be written without reference to matrices as

$$\frac{\partial(f \circ \mathbf{g})}{\partial x_j} = \sum_{k=1}^{p} \frac{\partial f}{\partial y_k}\frac{\partial g_k}{\partial x_j}; \tag{17.1.5}$$

with $1 \leq j \leq n$; each $\partial f/\partial y_k$ on the right side is evaluated at $\mathbf{g}(\mathbf{x})$, and the left side is evaluated at $\mathbf{x}$.

Where does the right side of (17.1.5) come from? It is the expression for the $1j$-entry in the $1 \times n$ product matrix $(Df)(\mathbf{g}(\mathbf{x}))\,(D\mathbf{g})(\mathbf{x})$ (we say more about this in Remark 17.4.1). A convenient way to think about (17.1.5) is that by writing $\mathbf{y} = \mathbf{g}(\mathbf{x})$, it expresses rates of change of f in terms of the x's as a sum of contributions of rates of change of f in terms of the y's multiplied by rates of change of the y's in terms of the x's: abbreviating the notation in (17.1.5) by writing f instead of $f \circ \mathbf{g}$ on the left side and writing y_k instead of g_k on the right side expresses it as:

$$\frac{\partial f}{\partial x_j} = \sum_{k=1}^{p} \frac{\partial f}{\partial y_k}\frac{\partial y_k}{\partial x_j} \tag{17.1.6}$$

(where the left side is evaluated at $\mathbf{x}$ and each $\partial f/\partial y_k$ on the right side is evaluated at $\mathbf{g}(\mathbf{x})$). Although (17.1.5) is the more precise formulation, **you will often encounter the notationally convenient version (17.1.6) in many places**. Observe that when $n = 1$ this recovers (17.1.4).

An important special case is when $n = p$; in this case often $\mathbf{x}$ and $\mathbf{y}$ are different coordinates on the same region of $\mathbf{R}^n$. The best-known example of this is the use of polar coordinates on $\mathbf{R}^2$, where $\mathbf{g}(r,\theta) = (r\cos\theta, r\sin\theta)$: we are "locating" a point on the plane in terms of its distance r from the origin and its angle θ from the positive x-axis (instead of its rectangular coordinates). Analogues with spherical and cylindrical coordinates for $n = 3$ arise in many physical problems.

17.2. Sample calculations. To get comfortable with the mechanics of this new Chain Rule, let's carry out some calculations with it. Some of the following examples use the more fundamental formulation in terms of matrix multiplication, and others use the more explicit scalar-valued formulas in (17.1.4), (17.1.5), and (17.1.6). Both versions of the Chain Rule are useful in practice.

Example 17.2.1. Consider the function $f : \mathbf{R}^3 \to \mathbf{R}^2$ given by

$$f(x,y,z) = (xy + (1/2)z^2, xz + (1/2)y^2)$$

and define $F(s,t) = f(t^2, st, 1/s)$; i.e., $F = f \circ g$ for $g(s,t) = (t^2, st, 1/s)$. Let's calculate $(DF)(1,2)$ in two ways: using the Chain Rule and using explicit calculation of the component functions of F.

Method 1. By the Chain Rule, $(DF)(1,2) = (Df)(g(1,2))(Dg)(1,2) = (Df)(4,2,1)(Dg)(1,2)$. Computing partial derivatives of component functions of f and g yields

$$(Df)(x,y,z) = \begin{bmatrix} y & x & z \\ z & y & x \end{bmatrix}, \quad (Dg)(s,t) = \begin{bmatrix} 0 & 2t \\ t & s \\ -1/s^2 & 0 \end{bmatrix}.$$

Hence,

$$(DF)(1,2) = (Df)(4,2,1)(Dg)(1,2) = \begin{bmatrix} 2 & 4 & 1 \\ 1 & 2 & 4 \end{bmatrix}\begin{bmatrix} 0 & 4 \\ 2 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 7 & 12 \\ 0 & 6 \end{bmatrix}.$$

Method 2. For the explicit approach, we compute

$$F(s,t) = (t^2(st) + (1/2)(1/s)^2, t^2(1/s) + (1/2)(st)^2) = (st^3 + (1/2)s^{-2}, t^2/s + s^2t^2/2),$$

so

$$(DF)(s,t) = \begin{bmatrix} t^3 - 1/s^3 & 3st^2 \\ -(t/s)^2 + st^2 & 2t/s + s^2 t \end{bmatrix}.$$

Thus,

$$(DF)(1,2) = \begin{bmatrix} 7 & 12 \\ 0 & 6 \end{bmatrix}.$$

■

Example 17.2.2. Suppose the temperature in a room is given by some function $T(x,y,z)$ at the point (x,y,z). A bug that begins at rest on the floor at $(4,2,0)$ flies around along the spiral path $p(t) = (3 + \cos t, 2 + \sin t, t)$ where t is time (measured in seconds). At time $t = 3$, what is the rate of change with respect to time for the temperature as experienced by the bug along its path of motion?

The temperature felt by the bug at time t is $f(t) = T(p(t)) = T(3 + \cos t, 2 + \sin t, t)$. If we write $p(t) = (x(t), y(t), z(t))$ then the Chain Rule in the form (17.1.4) says

$$f'(t) = \frac{\partial T}{\partial x}\frac{dx}{dt} + \frac{\partial T}{\partial y}\frac{dy}{dt} + \frac{\partial T}{\partial z}\frac{dz}{dt} = -(\sin t)\frac{\partial T}{\partial x} + (\cos t)\frac{\partial T}{\partial y} + \frac{\partial T}{\partial z}, \tag{17.2.1}$$

where it is understood that all partials of T are evaluated at the point $p(t) \in \mathbf{R}^3$.

For instance, if $T(x, y, z) = e^{-(x-2)^2-(y-2)^2} + z$ then

$$\frac{\partial T}{\partial x} = -2(x-2)e^{-(x-2)^2-(y-2)^2}, \quad \frac{\partial T}{\partial y} = -2(y-2)e^{-(x-2)^2-(y-2)^2}, \quad \frac{\partial T}{\partial z} = 1,$$

so evaluating these partials at $p(t) = (3 + \cos t, 2 + \sin t, t)$ gives

$$\frac{\partial T}{\partial x}(p(t)) = -2(1+\cos t)e^{-(1+\cos t)^2-(\sin t)^2}, \quad \frac{\partial T}{\partial y}(p(t)) = -2(\sin t)e^{-(1+\cos t)^2-(\sin t)^2}, \quad \frac{\partial T}{\partial z}(p(t)) = 1.$$

Plugging these into (17.2.1) and simplifying the algebra a bit, we get

$$f'(t) = 2(\sin t)e^{-(1+\cos t)^2-(\sin t)^2} + 1.$$

■

Example 17.2.3. Suppose $\mathbf{g} : \mathbf{R}^3 \to \mathbf{R}^2$ is

$$\mathbf{g}(x, y, z) = \begin{bmatrix} xy + z^2 \\ ze^y \end{bmatrix},$$

and $\mathbf{f} : \mathbf{R}^2 \to \mathbf{R}^2$ is $\mathbf{f}(v, w) = (v/w, v + w)$. The composition $\mathbf{h} = \mathbf{f} \circ \mathbf{g} : \mathbf{R}^3 \to \mathbf{R}^2$ carries (x, y, z) to

$$\begin{bmatrix} (xy + z^2)/(ze^y) \\ xy + z^2 + ze^y \end{bmatrix}.$$

In other words, if we write $\mathbf{h}$ in terms of component functions as $(h_1, h_2) : \mathbf{R}^3 \to \mathbf{R}^2$ for $h_1, h_2 : \mathbf{R}^3 \to \mathbf{R}$ then

$$h_1(x, y, z) = \frac{xy + z^2}{ze^y}, \quad h_2(x, y, z) = xy + z^2 + ze^y.$$

Computing the matrix $(D\mathbf{h})(1, 2, -1)$ of values of partial derivatives at $(1, 2, -1)$ of the component functions of $\mathbf{h}$ *directly* (i.e., without the Chain Rule) can be done in this case without too much fuss upon noting that we can write h_1 in the form $(xy/z)e^{-y} + ze^{-y}$, yielding

$$\frac{\partial h_1}{\partial x} = (y/z)e^{-y}, \quad \frac{\partial h_1}{\partial y} = ((x/z)(1-y) - z)e^{-y}, \quad \frac{\partial h_1}{\partial z} = (-(xy/z^2) + 1)e^{-y}.$$

Thus,

$$\frac{\partial h_1}{\partial x}(1, 2, -1) = -2e^{-2}, \quad \frac{\partial h_1}{\partial y}(1, 2, -1) = 2e^{-2}, \quad \frac{\partial h_1}{\partial z}(1, 2, -1) = -e^{-2}. \tag{17.2.2}$$

The partials of h_2 are $\partial h_2/\partial x = y$, $\partial h_2/\partial y = x + ze^y$, $\partial h_2/\partial z = 2z + e^y$, so evaluating at $(1, 2, -1)$ gives $(h_2)_x(1, 2, -1) = 2$, $(h_2)_y(1, 2, -1) = 1 - e^2$, $(h_2)_z(1, 2, -1) = -2 + e^2$. If the first component of $\mathbf{f}$ had been w/v rather than v/w then the first component h_1 of $\mathbf{h}$ would have been $(ze^y)/(xy + z^2)$ which does *not* break up as a sum, making direct computation of the partials of h_1 a bit of a mess in that case.

The Chain Rule provides a *completely different* way to compute those partials, without needing to compute h_1 and h_2 explicitly first. It enables us to efficiently organize the division of labor for computing partials of $\mathbf{h}$ into separately computing partials of $\mathbf{f}$ (without the intervention of $\mathbf{g}$!) and partials of $\mathbf{g}$. In cases where the component functions of $\mathbf{h}$ are complicated, this can be a real simplification.

Even in cases (as at present) where one *could* compute the partials of the component functions of $\mathbf{h} = \mathbf{f} \circ \mathbf{g}$ directly, the Chain Rule typically lets us get by with simpler calculations at every step (except we do have to carry out one matrix multiplication at the end). This was already seen when fleshing out Example 17.1.2 in Section 17.1.

Turning now to the Chain Rule in our setting above, we compute

$$(D\mathbf{f})(v,w) = \begin{bmatrix} 1/w & -v/w^2 \\ 1 & 1 \end{bmatrix}, \quad (D\mathbf{g})(x,y,z) = \begin{bmatrix} y & x & 2z \\ 0 & ze^y & e^y \end{bmatrix}$$

and $\mathbf{g}(1,2,-1) = (3,-e^2)$, so by the Chain Rule

$$\begin{aligned} (D\mathbf{h})(1,2,-1) = (D\mathbf{f})(3,-e^2)\,(D\mathbf{g})(1,2,-1) &= \begin{bmatrix} -1/e^2 & -3/e^4 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -2 \\ 0 & -e^2 & e^2 \end{bmatrix} \\ &= \begin{bmatrix} -2/e^2 & 2/e^2 & -1/e^2 \\ 2 & 1-e^2 & -2+e^2 \end{bmatrix}. \end{aligned}$$

The first row of the final matrix gives the partial derivatives of h_1 at $(1,2,-1)$ with respect to x,y,z in order (recovering what we computed directly in (17.2.2)), and likewise with the second row for h_2. ■

Example 17.2.4. Consider the composition $\mathbf{h} = \mathbf{f} \circ \mathbf{g} : \mathbf{R}^3 \to \mathbf{R}^3$ of functions $\mathbf{f} : \mathbf{R}^2 \to \mathbf{R}^3$ and $\mathbf{g} : \mathbf{R}^3 \to \mathbf{R}^2$ given by

$$\mathbf{f}(x,y) = (x^2, xy, y^2), \quad \mathbf{g}(r,s,t) = (r\sin(\pi s), -s+t).$$

In this case it actually *makes sense* to form the composite functions

$$\mathbf{h} = \mathbf{f} \circ \mathbf{g} : \mathbf{R}^3 \to \mathbf{R}^3, \quad \mathbf{H} = \mathbf{g} \circ \mathbf{f} : \mathbf{R}^2 \to \mathbf{R}^2$$

in *both* orders, but these are totally different things:

$$\mathbf{h}(r,s,t) = (r^2\sin^2(\pi s), r\sin(\pi s)(-s+t), (-s+t)^2), \quad \mathbf{H}(x,y) = (x^2\sin(\pi xy), -xy+y^2).$$

Let's compute $(D\mathbf{h})(1,1,-1)$ using the Chain Rule. By computing partial derivatives of component functions of $\mathbf{f}$ and $\mathbf{g}$ we have

$$(D\mathbf{f})(x,y) = \begin{bmatrix} 2x & 0 \\ y & x \\ 0 & 2y \end{bmatrix}, \quad (D\mathbf{g})(r,s,t) = \begin{bmatrix} \sin(\pi s) & \pi r\cos(\pi s) & 0 \\ 0 & -1 & 1 \end{bmatrix},$$

so since $\mathbf{g}(1,1,-1) = (0,-2)$ we have

$$(D\mathbf{h})(1,1,-1) = (D\mathbf{f})(0,-2)\,(D\mathbf{g})(1,1,-1) = \begin{bmatrix} 0 & 0 \\ -2 & 0 \\ 0 & -4 \end{bmatrix} \begin{bmatrix} 0 & -\pi & 0 \\ 0 & -1 & 1 \end{bmatrix}.$$

Multiplying these matrices gives the answer

$$(D\mathbf{h})(1,1,-1) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2\pi & 0 \\ 0 & 4 & -4 \end{bmatrix}.$$

As a safety check, the first two entries in the bottom row say the third component function $h_3(r,s,t) = (-s+t)^2$ of $\mathbf{h}$ satisfies $(\partial h_3/\partial r)(1,1,-1) = 0$ and $(\partial h_3/\partial s)(1,1,-1) = 4$, both of which can be checked by direct calculation with the formula for h_3.

Let's now do similarly for the composition $\mathbf{H}$ in the opposite order, and compute $(D\mathbf{H})(0,-2)$. Note that $\mathbf{f}(0,-2) = (0,0,4)$ (not $(1,1,-1)$!!), so by the Chain Rule

$$(D\mathbf{H})(0,-2) = (D\mathbf{g})(0,0,4)\,(D\mathbf{f})(0,-2) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ -2 & 0 \\ 0 & -4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 2 & -4 \end{bmatrix}. \tag{17.2.3}$$

For example, writing H_1 and H_2 for the component functions of $\mathbf{H} : \mathbf{R}^2 \to \mathbf{R}^2$, the right column of the final 2×2 matrix says $(\partial H_1/\partial y)(0,-2) = 0$ and $(\partial H_2/\partial y)(0,-2) = -4$, as can be verified directly from the formulas $H_1(x,y) = x^2 \sin(\pi xy)$ and $H_2(x,y) = -xy + y^2$.

Beware that although in this case it also makes sense algebraically to form the product matrix

$$(D\mathbf{g})(1,1,-1)\,(D\mathbf{f})(0,-2)$$

in the opposite order, this has *nothing* to do with $(D(\mathbf{g} \circ \mathbf{f}))(0,-2) = (D\mathbf{H})(0,-2)$. Indeed, the latter was computed in (17.2.3), and is not equal to

$$(D\mathbf{g})(1,1,-1)\,(D\mathbf{f})(0,-2) = \begin{bmatrix} 0 & -\pi & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ -2 & 0 \\ 0 & -4 \end{bmatrix} = \begin{bmatrix} 2\pi & 0 \\ 2 & -4 \end{bmatrix}$$

(the upper-left entries are different). This does *not* contradict the Chain Rule for $\mathbf{H}$ at $(0,2)$ since that instance of the Chain Rule requires evaluating $D\mathbf{g}$ at the point $\mathbf{f}(0,2) = (0,0,4)$ rather than at $(1,1,-1)$. ■

The preceding example illustrates a point to keep in mind:

> The order of multiplication of matrices in the Chain Rule matches the order of composition of the functions being composed. Even if it "makes sense" to multiply the matrices in the wrong order, the opposite-order product is generally *not* the derivative matrix for the opposite-order composite function (since evaluation is usually at the wrong points). Be careful about the order of matrix multiplication!

Example 17.2.5. In single-variable calculus, we learn the "product rule" for derivatives:

$$(g_1 g_2)'(x) = g_1'(x)g_2(x) + g_1(x)g_2'(x).$$

This fact from single-variable calculus has a striking explanation in terms of the *multivariable* Chain Rule, as follows. The idea is to express the product function $g_1 g_2 : \mathbf{R} \to \mathbf{R}$ in the single-variable setting as a *multivariable* composition $\mathbf{f} \circ \mathbf{g}$ according to the picture

$$\mathbf{R} \xrightarrow{\mathbf{g}} \mathbf{R}^2 \xrightarrow{\mathbf{f}} \mathbf{R}$$

where $\mathbf{g}(x) = (g_1(x), g_2(x))$ and $\mathbf{f}(y_1, y_2) = y_1 y_2$ (so $\mathbf{f}$ coincides with M in Example 8.1.4). Check for yourself that indeed $\mathbf{f} \circ \mathbf{g}$ is the function with value $g_1(x)g_2(x)$ at x.

This composition "decouples" the multiplication step from the consideration of the two separate functions g_1 and g_2. By the rules for derivative matrices in terms of partial derivatives, we have

$$(D\mathbf{f})(y_1, y_2) = [y_2 \; y_1], \quad (D\mathbf{g})(x) = \begin{bmatrix} g_1'(x) \\ g_2'(x) \end{bmatrix},$$

so upon multiplying these *in the correct order* we see that the 1×1 matrix $[(g_1 g_2)'(x)]$ is the product

$$(D\mathbf{f})(\mathbf{g}(x))\,(D\mathbf{g})(x) = (D\mathbf{f})(g_1(x), g_2(x))\,(D\mathbf{g})(x) = [g_2(x) \; g_1(x)] \begin{bmatrix} g_1'(x) \\ g_2'(x) \end{bmatrix},$$

and by the rules for matrix multiplication this is the 1×1 matrix $[g_2(x)g_1'(x) + g_1(x)g_2'(x)]$. Staring at the single entry in this matrix, we get the old formula for $(g_1 g_2)'(x)$. This is a totally new explanation of the product rule from *single-variable* calculus, by using the *multivariable* Chain Rule! ■

In Example 17.2.6 below we will see that expressing a single-variable function as a multivariable composition naturally arises in the context of computing the velocity for a comet orbiting a star. There is an abundance of other applications of the same idea (often when seeking to compute rates of change with respect to time), some of which were mentioned at the start of this chapter.

Example 17.2.6. The position of a comet orbiting a star at $(0,0)$ on the xy-plane is given by the function $\mathbf{p}(t) = (3\cos t + 1, \sin t)$ graphed below:

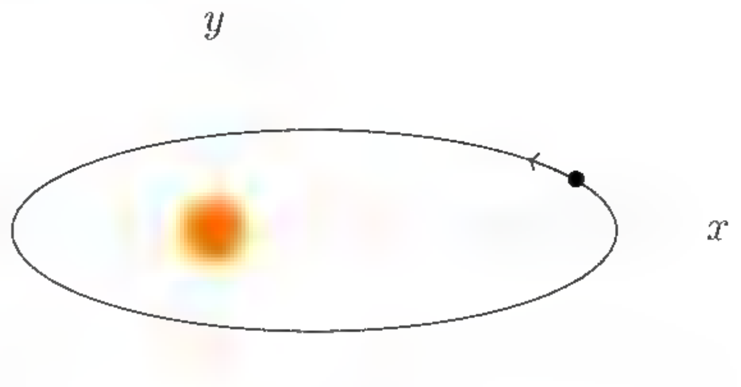


FIGURE 17.2.1. A comet orbiting a star at the origin

(The output of the function $\mathbf{p} : \mathbf{R} \to \mathbf{R}^2$ consists of the points on an ellipse with $(0,0)$ as one of its foci, in accordance with Kepler's Laws, but that is not needed in the calculations which follow.)

Suppose we want to find the points along the orbit where the comet is closest or furthest from the star. This amounts to finding the maxima and minima of the distance $d(t)$ between the comet and the star at time t. Those maxima and minima occur at those times t_0 satisfying $d'(t_0) = 0$. For this reason, it is of interest to an astronomer to compute the derivative $d'(t)$. (Note that since $d(t) \geq 0$, the maxima and minima of d are the same as for d^2, so for the purpose of finding the times at which the extreme values are attained we could replace d with d^2. This has the merit of removing some annoying square roots which otherwise arise from the Pythagorean Theorem below. We make the direct calculations below without that trick, since our aim is to illustrate the consistency of the Chain Rule with other methods of calculation, and it comes out more impressively by retaining the square roots below.)

Note that the distance d is very naturally a composite function:

$$d(t) = \sqrt{(3\cos t + 1)^2 + (\sin t)^2} = (F \circ \mathbf{p})(t)$$

where $F(x,y) = \sqrt{x^2 + y^2}$. We shall now compute d' in two different ways: first we will find a formula for the single-variable function $d(t)$ and compute its derivative using single-variable calculus, and then we will use the Chain Rule via the above expression for $d : \mathbf{R} \to \mathbf{R}$ as a composition $\mathbf{R} \to \mathbf{R}^2 \to \mathbf{R}$ of *multivariable* functions.

The first method below looks shorter only because it is a more familiar technique (from single-variable calculus) and we have done the messy algebra for you; it is not really simpler. After reading both methods, you should recognize that the second method is not longer after one gets accustomed to the Chain Rule.

- (First method) Since

$$d(t) = \sqrt{(3\cos t + 1)^2 + (\sin t)^2} = \sqrt{9\cos^2 t + 6\cos t + \sin^2 t + 1},$$

taking the derivative with respect to t (using the single-variable Chain Rule) yields

$$\begin{aligned} d'(t) &= \left(\sqrt{9\cos^2 t + 6\cos t + \sin^2 t + 1}\right)' \\ &= \frac{(9\cos^2 t + 6\cos t + \sin^2 t + 1)'}{2\sqrt{9\cos^2 t + 6\cos t + \sin^2 t + 1}} \\ &= \frac{-18\cos t \sin t - 6\sin t + 2\sin t\cos t}{2\sqrt{9\cos^2 t + 6\cos t + \sin^2 t + 1}} \\ &= -\frac{8\sin t\cos t + 3\sin t}{\sqrt{9\cos^2 t + 6\cos t + \sin^2 t + 1}}. \end{aligned}$$

- (Second method) Defining $x(t) = 3\cos t + 1$ and $y(t) = \sin t$ to be the components of $\mathbf{p}(t)$, for $d(t) = (F \circ \mathbf{p})(t)$ we get from (17.1.4)
$$d'(t) = \frac{\partial F}{\partial x}\frac{dx}{dt} + \frac{\partial F}{\partial y}\frac{dy}{dt} = (-3\sin t)\frac{\partial F}{\partial x} + (\cos t)\frac{\partial F}{\partial y}.$$
The partials of F are
$$\frac{\partial}{\partial x}(\sqrt{x^2+y^2}) = \frac{x}{\sqrt{x^2+y^2}}, \quad \frac{\partial}{\partial y}(\sqrt{x^2+y^2}) = \frac{y}{\sqrt{x^2+y^2}},$$
to be evaluated at $\mathbf{p}(t) = (x(t), y(t))$. Hence,
$$\begin{aligned} d'(t) &= (-3\sin t)\frac{x(t)}{\sqrt{x(t)^2+y(t)^2}} + (\cos t)\frac{y(t)}{\sqrt{x(t)^2+y(t)^2}} \\ &= (-3\sin t)\frac{3\cos t+1}{\sqrt{(3\cos t+1)^2+(\sin t)^2}} + (\cos t)\frac{\sin t}{\sqrt{(3\cos t+1)^2+(\sin t)^2}} \\ &= -\frac{8\sin t\cos t + 3\sin t}{\sqrt{(3\cos t+1)^2+\sin^2 t}}. \end{aligned}$$

Though these two methods of computing $d'(t)$ are very different, they give the same answer (as they must if we didn't make a mistake): the numerators in both fractional expressions are literally the same, and the expressions inside the square roots in the denominators for the two fractional expressions for $d'(t)$ are seen to be equal upon expanding out $(3\cos t + 1)^2 = 9\cos^2 t + 6\cos t + 1$. ■

Remark 17.2.7. Using the formula for $d'(t)$ and considering where it vanishes, we can find the critical points of $d(t)$. Namely, they correspond to where $\sin t = 0$ (so $\cos t = \pm 1$) or $\cos t = -3/8$ (so $\sin t = \pm\sqrt{55}/8$). The positions $\mathbf{p}(t) = (3\cos t + 1, \sin t)$ at such times are respectively $(4, 0), (-2, 0)$ and $(-1/8, \pm\sqrt{55}/8)$. These points have respective distances from (0,0) given by 4, 2, and $\sqrt{7/8} = 0.935\ldots$.

Thus, the point at greatest distance from the sun is $(4, 0)$, as is obvious from the picture of the orbit in Figure 17.2.1, and the points at closest distance to the sun are $(-1/8, \pm\sqrt{55}/8)$. The latter is not easy to see from Figure 17.2.1, but it is clear from the picture that $(-2, 0)$ at the opposite end of the orbit from $(4, 0)$ is definitely not closest to the sun and the points that are closest are a pair a bit to the left of the y-axis (as a safety check on the reasonableness of our determination of those points as $(-1/8, \pm\sqrt{55}/8)$).

In Examples 17.2.2 and 17.2.6 we saw the scalar-valued (17.1.4) in action. The next example illustrates its generalizations (17.1.5) and (17.1.6) for situations when the source involves more than one variable.

Example 17.2.8. Consider a function $f(x, y, z)$ whose variables x, y, z each depend on another pair of variables u, w via the formulas
$$x(u, w) = \sin(u)w, \quad y(u, w) = w^2, \quad z(u, w) = u/w. \tag{17.2.4}$$
Via these dependencies, f can be viewed as a function of (u, w), namely $f(\sin(u)w, w^2, u/w)$. We shall compute the partials of f with respect to u and w in terms of its partials with respect to x, y, z.

The general formula (17.1.5) (expressed as in (17.1.6)) gives
$$\frac{\partial f}{\partial u} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial u} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial u} + \frac{\partial f}{\partial z}\frac{\partial z}{\partial u}. \tag{17.2.5}$$
We compute the partials of x, y, z with respect to u using (17.2.4):
$$\frac{\partial x}{\partial u} = \cos(u)w, \quad \frac{\partial y}{\partial u} = 0, \quad \frac{\partial z}{\partial u} = \frac{1}{w}.$$

Plugging these into the right side of (17.2.5) yields

$$\frac{\partial f}{\partial u} = \cos(u) w \frac{\partial f}{\partial x} + \frac{1}{w}\frac{\partial f}{\partial z}. \tag{17.2.6}$$

For example, at $(u, w) = (0, -2)$ we have $(x, y, z) = (0, 4, 0)$ due to (17.2.4), so (17.2.6) gives

$$\frac{\partial f}{\partial u}(0, -2) = -2\frac{\partial f}{\partial x}(0, 4, 0) - \frac{1}{2}\frac{\partial f}{\partial z}(0, 4, 0).$$

Likewise, to compute $\partial f/\partial w$ we use the general formula (17.1.5) (expressed as in (17.1.6)) to get

$$\frac{\partial f}{\partial w} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial w} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial w} + \frac{\partial f}{\partial z}\frac{\partial z}{\partial w}. \tag{17.2.7}$$

We compute the partials of x, y, z with respect to w using (17.2.4):

$$\frac{\partial x}{\partial w} = \sin(u), \quad \frac{\partial y}{\partial w} = 2w, \quad \frac{\partial z}{\partial w} = -\frac{u}{w^2}.$$

Plugging these into the right side of (17.2.7) yields

$$\frac{\partial f}{\partial w} = \sin(u)\frac{\partial f}{\partial x} + 2w\frac{\partial f}{\partial y} - \frac{u}{w^2}\frac{\partial f}{\partial z}. \tag{17.2.8}$$

For example, at $(u, w) = (\pi/2, -2)$ we have $(x, y, z) = (-2, 4, -\pi/4)$ due to (17.2.4), so

$$\frac{\partial f}{\partial w}(\pi/2, -2) = \frac{\partial f}{\partial x}(-2, 4, -\pi/4) - 4\frac{\partial f}{\partial y}(-2, 4, -\pi/4) - \frac{\pi}{8}\frac{\partial f}{\partial z}(-2, 4, -\pi/4). \qquad \blacksquare$$

Example 17.2.9. Let's push the calculations in Example 17.2.8 further: suppose u and w depend on another variable r, say

$$u = r^3, \quad w = e^r.$$

Thus, we can view f as a function of r, namely

$$f(\sin(u)w, w^2, u/w)|_{(u,w)=(r^3,e^r)} = f(\sin(r^3)e^r, e^{2r}, r^3e^{-r}).$$

We will compute $df/dr = \partial f/\partial r$ in *three ways* in terms of the partials of f with respect to x, y, z. These approaches will all give the same answer, as they must if we don't make a mistake. (In terms of the matrix Chain Rule, the equality of the three approaches below expresses the associativity of matrix multiplication, but we omit discussion of that and instead just work out the calculations that illustrate the consistency.)

First, using (17.1.4) going through the dependence of f on (u, v) which depends on r gives

$$\frac{df}{dr} = \frac{\partial f}{\partial u}\frac{du}{dr} + \frac{\partial f}{\partial w}\frac{dw}{dr} = (3r^2)\frac{\partial f}{\partial u} + e^r\frac{\partial f}{\partial w}.$$

Then we plug in (17.2.6) and (17.2.8) with coefficients evaluated at $(u, w) = (r^3, e^r)$ to get

$$\frac{df}{dr} = (3r^2)(\cos(r^3)e^r\frac{\partial f}{\partial x} + e^{-r}\frac{\partial f}{\partial z}) + e^r(\sin(r^3)\frac{\partial f}{\partial x} + 2e^r\frac{\partial f}{\partial y} - r^3e^{-2r}\frac{\partial f}{\partial z}).$$

Collecting the occurrences of $\partial f/\partial x$, $\partial f/\partial y$, and $\partial f/\partial z$ simplifies this to:

$$\frac{df}{dr} = (3\cos(r^3)r^2 + \sin(r^3))e^r\frac{\partial f}{\partial x} + 2e^{2r}\frac{\partial f}{\partial y} + (3r^2 - r^3)e^{-r}\frac{\partial f}{\partial z}. \tag{17.2.9}$$

Here is another approach: using (17.1.4) going through the dependence of f on (x, y, z) which depends on r (via $x(u, w) = x(r^3, e^r)$, $y(u, w) = y(r^3, e^r)$, $z(u, w) = z(r^3, e^r)$) gives

$$\frac{df}{dr} = \frac{\partial f}{\partial x}\frac{dx}{dr} + \frac{\partial f}{\partial y}\frac{dy}{dr} + \frac{\partial f}{\partial z}\frac{dz}{dr}. \tag{17.2.10}$$

We next apply (17.1.4) to each of x, y, z as a function of r through (u, v) using (17.2.4):

$$\begin{aligned}
\frac{dx}{dr} &= \frac{\partial x}{\partial u}\frac{du}{dr} + \frac{\partial x}{\partial w}\frac{dw}{dr} = \cos(u) w\, u'(r) + \sin(u) w'(r) = \cos(r^3)e^r(3r^2) + \sin(r^3)e^r,\\
\frac{dy}{dr} &= \frac{\partial y}{\partial u}\frac{du}{dr} + \frac{\partial y}{\partial w}\frac{dw}{dr} = 0\, u'(r) + 2w\, w'(r) = 2e^r e^r = 2e^{2r},\\
\frac{dz}{dr} &= \frac{\partial z}{\partial u}\frac{du}{dr} + \frac{\partial z}{\partial w}\frac{dw}{dr} = \frac{1}{w}u'(r) - \frac{u}{w^2}w'(r) = e^{-r}(3r^2) - \frac{r^3}{e^{2r}}w'(r) = 3r^2e^{-r} - r^3e^{-r}.
\end{aligned}$$

Plugging these into (17.2.10), we get *exactly* the expression (17.2.9).

And for yet another approach (which is very close to the second approach above), f as a function of r is obtained from (17.2.4) via the expressions $x(r^3, e^r) = \sin(r^3)e^r$, $y(r^3, e^r) = e^{2r}$, and $z(r^3, e^r) = r^3e^{-r}$: the function is $f(\sin(r^3)e^r, e^{2r}, r^3e^{-r})$. The derivative of this with respect to r is given by (17.1.4):

$$\frac{df}{dr} = \frac{\partial f}{\partial x}\frac{d}{dr}(\sin(r^3)e^r) + \frac{\partial f}{\partial y}\frac{d}{dr}(e^{2r}) + \frac{\partial f}{\partial z}\frac{d}{dr}(r^3e^{-r})$$

(this is essentially (17.2.10) with explicit expressions in r substituted for each of x, y, z). Working out these r-derivatives explicitly again yields (17.2.9). ■

17.3. Evaluating at numerical points. When evaluating the derivative $D(\mathbf{f} \circ \mathbf{g})$ *at a specific point* $\mathbf{a} \in \mathbf{R}^n$, we can proceed in one of two ways:

(a) Compute the matrices $(D\mathbf{f})(\mathbf{g}(\mathbf{x}))$ and $(D\mathbf{g})(\mathbf{x})$ whose entries may contain the *variables* $x_1, \dots, x_n$, multiply them together, and then plug in $\mathbf{x} = \mathbf{a}$ at the end.
(b) Compute $(D\mathbf{f})(\mathbf{y})$ and $(D\mathbf{g})(\mathbf{x})$, plug in $\mathbf{y} = \mathbf{g}(\mathbf{a})$ and $\mathbf{x} = \mathbf{a}$ to obtain matrices whose entries are *numbers*, and then multiply those numerical matrices together.

The latter option is sometimes easier in practice because multiplying matrices containing variables may lead to algebraic tedium or errors whereas using specific numbers can make the matrices simpler for multiplication purposes (though that is not always the case: sometimes working with general formulas, when not too messy, conveys the structure of a calculation more clearly). To summarize:

When evaluating a composite derivative matrix at a numerical point, it may be simpler to evaluate both ingredients in the Chain Rule at the corresponding numerical points before multiplying the matrices.

This is illustrated the next two examples, the second of which is perhaps a more dramatic illustration than the first of how (b) can be much cleaner than (a).

Example 17.3.1. Let $g : \mathbf{R}^2 \to \mathbf{R}^3$ be the function $g(s,t) = \begin{bmatrix} t^2 \\ st \\ 1/s \end{bmatrix}$ and let $f : \mathbf{R}^3 \to \mathbf{R}^2$ be the function

$$f(x,y,z) = \begin{bmatrix} (1/2)(x^2+y^2+z^2) \\ xz + (1/2)y^2 \end{bmatrix}.$$

For the composite function $f \circ g : \mathbf{R}^2 \to \mathbf{R}^2$, we will compute $(D(f \circ g))(1,2)$ in three different ways:

(a) we evaluate $(D(f \circ g))(s,t) = (Df)(g(s,t))\,(Dg)(s,t)$ by computing $(Df)(x,y)$ and $(Dg)(s,t)$ in terms of variables and multiplying the matrices, and at the end setting $s = 1$ and $t = 2$,
(b) we evaluate $(D(f \circ g))(1,2) = (Df)(g(1,2))\,(Dg)(1,2)$ by computing both matrices on the right side in terms of numbers, and then multiplying the matrices,
(c) we compute $f(g(s,t))$ explicitly and evaluate the partial derivatives of its component functions at the point $(1,2)$.

In approaches (a) and (b), we first need to compute partial derivatives of the component functions of f and of g to get

$$(Df)(x,y,z) = \begin{bmatrix} x & y & z \\ z & y & x \end{bmatrix}, \quad (Dg)(s,t) = \begin{bmatrix} 0 & 2t \\ t & s \\ -1/s^2 & 0 \end{bmatrix}.$$

For approach (a), we compute symbolically that

$$(Df)(g(s,t)) = \begin{bmatrix} t^2 & st & 1/s \\ 1/s & st & t^2 \end{bmatrix},$$

so (using our above determination of $(Dg)(s,t)$)

$$(Df)(g(s,t))\,(Dg)(s,t) = \begin{bmatrix} t^2 & st & 1/s \\ 1/s & st & t^2 \end{bmatrix} \begin{bmatrix} 0 & 2t \\ t & s \\ -1/s^2 & 0 \end{bmatrix} = \begin{bmatrix} st^2 - 1/s^3 & 2t^3 + s^2t \\ st^2 - t^2/s^2 & (2t/s) + s^2t \end{bmatrix}.$$

Finally substituting $s = 1$ and $t = 2$ yields

$$\begin{bmatrix} 3 & 18 \\ 0 & 6 \end{bmatrix}.$$

For approach (b), we numerically evaluate $(Df)(x,y,z)$ at $g(1,2) = (4,2,1)$ to get

$$(Df)(4,2,1) = \begin{bmatrix} 4 & 2 & 1 \\ 1 & 2 & 4 \end{bmatrix}, \quad (Dg)(1,2) = \begin{bmatrix} 0 & 4 \\ 2 & 1 \\ -1 & 0 \end{bmatrix}$$

(the latter a numerical evaluation of our general computation of $(Dg)(s,t)$). Multiplying these gives

$$\begin{bmatrix} 4 & 2 & 1 \\ 1 & 2 & 4 \end{bmatrix} \begin{bmatrix} 0 & 4 \\ 2 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 3 & 18 \\ 0 & 6 \end{bmatrix},$$

the same answer as via approach (a).

Finally, approach (c) is the "brute force" method which bypasses the Chain Rule in favor of a direct calculation. We have

$$f(g(s,t)) = f(t^2, st, 1/s) = ((1/2)(t^4 + s^2t^2 + 1/s^2), t^2/s + (1/2)s^2t^2).$$

Computing partial derivatives of the component functions yields

$$\begin{bmatrix} st^2 - 1/s^3 & 2t^3 + s^2t \\ -t^2/s^2 + st^2 & 2t/s + s^2t \end{bmatrix}.$$

This agrees with our computation of the matrix product $(Df)(g(s,t))\,(Dg)(s,t)$ in approach (a), as it must, and evaluating this at $(s,t) = (1,2)$ is exactly what we did at the end of approach (a). ■

Example 17.3.2. Let $f : \mathbf{R}^3 \to \mathbf{R}^2$ and $g : \mathbf{R}^4 \to \mathbf{R}^3$ be the functions

$$f(x,y,z) = \begin{bmatrix} x^2 - xy + y^2 \\ 4e^{x+y\quad z} \end{bmatrix}$$

and

$$g(q,r,s,t) = \begin{bmatrix} qr \\ rs \\ st \end{bmatrix}.$$

Consider the task of computing the derivative of $f \circ g : \mathbf{R}^4 \to \mathbf{R}^2$ at the point $(q, r, s, t) = (1, 2, 3, 4)$. Finding a *general* formula for $(Df) \circ g$ and multiplying it against a general formula for Dg in this case is a mess. (Please try it for yourself.) So rather than follow option (a) above, we will follow option (b).

First, we have

$$(Dg)(q, r, s, t) = \begin{bmatrix} r & q & 0 & 0 \\ 0 & s & r & 0 \\ 0 & 0 & t & s \end{bmatrix},$$

so

$$(Dg)(1, 2, 3, 4) = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 3 & 2 & 0 \\ 0 & 0 & 4 & 3 \end{bmatrix}.$$

We also have

$$\begin{aligned}(Df)(x, y, z) &= \begin{bmatrix} \frac{\partial}{\partial x}(x^2 - xy + y^2) & \frac{\partial}{\partial y}(x^2 - xy + y^2) & \frac{\partial}{\partial z}(x^2 - xy + y^2) \\ \frac{\partial}{\partial x}(4e^{x+y-z}) & \frac{\partial}{\partial y}(4e^{x+y-z}) & \frac{\partial}{\partial z}(4e^{x+y-z}) \end{bmatrix} \\ &= \begin{bmatrix} 2x - y & -x + 2y & 0 \\ 4e^{x+y-z} & 4e^{x+y-z} & -4e^{x+y-z} \end{bmatrix}\end{aligned}$$

and so

$$(Df)(g(1, 2, 3, 4)) = (Df)(2, 6, 12) = \begin{bmatrix} -2 & 10 & 0 \\ 4e^{-4} & 4e^{-4} & -4e^{-4} \end{bmatrix}.$$

The product of these matrices (in the correct order) is

$$\begin{bmatrix} -2 & 10 & 0 \\ 4e^{-4} & 4e^{-4} & -4e^{-4} \end{bmatrix} \begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 3 & 2 & 0 \\ 0 & 0 & 4 & 3 \end{bmatrix} = \begin{bmatrix} -4 & 28 & 20 & 0 \\ 8e^{-4} & 16e^{-4} & -8e^{-4} & -12e^{-4} \end{bmatrix}.$$

For instance, the composition $h = f \circ g : \mathbf{R}^4 \to \mathbf{R}^2$ has second component $h_2(q, r, s, t) = 4e^{qr+rs-st}$ and the third entry in the bottom row of the final 2×4 matrix says $(\partial h_2/\partial s)(1, 2, 3, 4) = -8e^{-4}$, as can also be verified directly from the formula for h_2 since $\partial h_2/\partial s = 4(r - t)e^{qr+rs-st}$. ■

17.4. Where does the multivariable Chain Rule come from? Consider functions $\mathbf{f} : \mathbf{R}^p \to \mathbf{R}^m$ and $\mathbf{g} : \mathbf{R}^n \to \mathbf{R}^p$, and a point $\mathbf{a} \in \mathbf{R}^n$. We shall now explain informally why the multivariable Chain Rule

$$(D(\mathbf{f} \circ \mathbf{g}))(\mathbf{a}) = (D\mathbf{f})(\mathbf{g}(\mathbf{a})) \cdot (D\mathbf{g})(\mathbf{a})$$

holds, expressing the derivative matrix of a composite function as the product of the derivative matrices of the functions being composed. Using the viewpoint of matrix multiplication as expressing a *composition* of linear transformations, this takes on the more appealing form: "derivative of a composition is the composition of the derivatives". To arrive at such a result, we will use the link between derivative matrices and good linear approximations.

In (13.5.3) we saw that the best linear approximation to the function $\mathbf{g}$ at the point $\mathbf{a}$ can be computed with the derivative matrix at $\mathbf{a}$. Informally in symbols:

$$\mathbf{g}(\mathbf{x}) \approx \mathbf{g}(\mathbf{a}) + ((D\mathbf{g})(\mathbf{a}))(\mathbf{x} - \mathbf{a})$$

for $\mathbf{x}$ near $\mathbf{a}$. To simplify the notation, let's write $L = (D\mathbf{g})(\mathbf{a})$ to denote the derivative matrix of $\mathbf{g}$ at the point $\mathbf{a}$. The best linear approximation to $\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{a})$ is then

$$\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{a}) \approx \underbrace{L(\mathbf{x} - \mathbf{a})}_{\text{matrix-vector mult.}}$$

for $\mathbf{x}$ near $\mathbf{a}$.

Also let us write $M = D\mathbf{f}(\mathbf{g}(\mathbf{a}))$ to denote the derivative matrix of $\mathbf{f}$ at the point $\mathbf{g}(\mathbf{a})$. Thus,

$$\mathbf{f}(\mathbf{y}) - \mathbf{f}(\mathbf{g}(\mathbf{a})) \approx \underbrace{M(\mathbf{y} - \mathbf{g}(\mathbf{a}))}_{\text{matrix-vector mult.}} \tag{17.4.1}$$

for $\mathbf{y}$ near $\mathbf{g}(\mathbf{a})$.

Now we put these two approximations together to study the composition $\mathbf{h} = \mathbf{f} \circ \mathbf{g}$. For $\mathbf{x}$ near $\mathbf{a}$ we have that $\mathbf{g}(\mathbf{x})$ is near $\mathbf{g}(\mathbf{a})$ and so (17.4.1) applies with $\mathbf{y} = \mathbf{g}(\mathbf{x})$ to give

$$\mathbf{f}(\mathbf{g}(\mathbf{x})) - \mathbf{f}(\mathbf{g}(\mathbf{a})) \approx M(\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{a})) \approx M(L(\mathbf{x} - \mathbf{a})) = (ML)(\mathbf{x} - \mathbf{a})$$

for $\mathbf{x}$ near $\mathbf{a}$. (Observe that at exactly this step, we have really been performing a *composition* of the best linear approximations; in the language of matrix-matrix products and matrix-vector products, it comes out as the matrix product ML.) This says

$$\mathbf{h}(\mathbf{x}) - \mathbf{h}(\mathbf{a}) \approx (ML)(\mathbf{x} - \mathbf{a})$$

for $\mathbf{x}$ near $\mathbf{a}$. Thus, ML gives a good linear approximation to $\mathbf{h}(\mathbf{x}) - \mathbf{h}(\mathbf{a})$ for $\mathbf{x}$ near $\mathbf{a}$. If one is careful about estimating the errors in these approximations, it can be shown that as $\mathbf{x}$ gets arbitrarily close to $\mathbf{a}$, the quality of ML as a linear approximation becomes so good that it is the "best" such linear approximation in the limit as $\mathbf{x}$ approaches $\mathbf{a}$, so it must be the derivative matrix $(D\mathbf{h})(\mathbf{a})$ for $\mathbf{h}$ at $\mathbf{a}$. But $ML = (D\mathbf{f})(\mathbf{g}(\mathbf{a}))\,(D\mathbf{g})(\mathbf{a})$, so we are done.

Remark 17.4.1. By using the formula for the entries of a derivative matrix and the formula for the entries of a matrix product, we can reformulate the Chain Rule more explicitly in the language of partial derivatives of component functions as in (17.1.5) or (17.1.6), as we now explain.

Write $\mathbf{g} : \mathbf{R}^n \to \mathbf{R}^p$ in terms of component functions as

$$\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_p(\mathbf{x}))$$

for $\mathbf{x} = (x_1, \dots, x_n) \in \mathbf{R}^n$. For $f : \mathbf{R}^p \to \mathbf{R}$, the composite function $h = f \circ \mathbf{g} : \mathbf{R}^n \to \mathbf{R}$ takes $\mathbf{x}$ to $f(\mathbf{g}(\mathbf{x}))$.

For $1 \le j \le n$, the $1 \times n$ matrix $(Dh)(\mathbf{x})$ (a single "row") has jth entry $\partial h/\partial x_j$, and this is equal to the $1j$-entry of $(Df)(\mathbf{g}(\mathbf{x}))\,(D\mathbf{g})(\mathbf{x})$ by the Chain Rule, where $(Df)(\mathbf{g}(\mathbf{x}))$ is also a single "row" (a $1 \times p$ matrix). Writing $\mathbf{y} = (y_1, \dots, y_p) \in \mathbf{R}^p$, the $1j$-entry of $(Df)(\mathbf{g}(\mathbf{x}))\,(D\mathbf{g})(\mathbf{x})$ is

$$\frac{\partial f}{\partial y_1}\frac{\partial g_1}{\partial x_j} + \frac{\partial f}{\partial y_2}\frac{\partial g_2}{\partial x_j} + \cdots + \frac{\partial f}{\partial y_p}\frac{\partial g_p}{\partial x_j}$$

where each $\partial f/\partial y_k$ is understood to be evaluated at $\mathbf{g}(\mathbf{x})$. Thus,

$$\frac{\partial h}{\partial x_j} = \sum_{k=1}^{p} \frac{\partial f}{\partial y_k}\frac{\partial g_k}{\partial x_j} \tag{17.4.2}$$

(where again each $\partial f/\partial y_k$ is evaluated at $\mathbf{g}(\mathbf{x})$), recovering (17.1.5) since $h = f \circ \mathbf{g}$.

Chapter 17 highlights (links to highlights in and next chapters)

Notation	**Meaning**	**Location in text**
nothing new!		

Concept	**Meaning**	**Location in text**
composition of multivariable functions	in many situations, a multivariable function has its variables all depending on some other collection of variables, and the cumulative dependence is a composition	Example 17.1.4
functions $f : \mathbf{R} \rightarrow \mathbf{R}$ sometimes can be usefully studied via multivariable methods	in many examples, f may be expressed as a composition $\mathbf{R} \rightarrow \mathbf{R}^n \rightarrow \mathbf{R}$ with $n > 1$	Ex. 17.1.2, 17.1.9, 17.2.2, 17.2.6
linear approximation of a composition is a matrix product	to linearly approximate a composition we linearly approximate each step, and then the original composition is approximated by a composition of linear functions, which in turn is given by a matrix product	discussion right after Theorem 17.1.5

Result	**Meaning**	**Location in text**
multivariable Chain Rule	the derivative matrix for $\mathbf{f} \circ \mathbf{g}$ at $\mathbf{a}$ is the product (in the same order) of the derivative matrix of $\mathbf{f}$ at $\mathbf{g}(\mathbf{a})$ and the derivative matrix of $\mathbf{g}$ at $\mathbf{a}$	Thm. 17.1.5, box above Ex. 17.1.8
dot product formula for derivative of composition $\mathbf{R} \rightarrow \mathbf{R}^p \rightarrow \mathbf{R}$	for $\mathbf{g} : \mathbf{R} \rightarrow \mathbf{R}^p$ and $f : \mathbf{R}^p \rightarrow \mathbf{R}$, the single-variable derivative of $f(\mathbf{g}(t))$ is $(\nabla f)(\mathbf{g}(t)) \cdot \mathbf{g}'(t)$	(17.1.3), (17.1.4)

Skill	**Location in text**
use matrix multiplication to compute the derivative matrix of a composition at a specific point (knowing at which point to evaluate each derivative matrix)	Examples 17.1.6, 17.1.7, 17.2.1, 17.2.3, 17.2.4
for composition $\mathbf{R} \rightarrow \mathbf{R}^p \rightarrow \mathbf{R}$, compute its single-variable derivative as dot product of gradient and a component-wise differentiated vector (or equivalently: (17.1.4))	Examples 17.1.7, 17.1.8, 17.2.2, 17.2.6
be aware of compatibility between symbolic and numerical uses of the multivariable Chain Rule	Section 17.3
for $\mathbf{g} : \mathbf{R}^n \rightarrow \mathbf{R}^p$ and $f : \mathbf{R}^p \rightarrow \mathbf{R}$ with coordinates $(x_1, \dots, x_n)$ on $\mathbf{R}^n$ and $(y_1, \dots, y_p)$ on $\mathbf{R}^p$, compute partials of "f in terms of x's" (i.e., partials of $f \circ \mathbf{g}$) in terms of partials of "f in terms of y's" and partials of component functions of $\mathbf{g}$	Remark 17.1.10, Exs. 17.2.8, 17.2.9

18. Matrix inverses and multivariable Newton's method for zeros

We have seen several operations on matrices – addition, subtraction, and multiplication – but the absence of division is notable. We now learn how to divide one matrix by another (but only very carefully, because *this operation only makes sense for certain matrices*). When working with nonzero scalars a, often $1/a$ is called the "reciprocal" of a; for matrices we use the word "inverse" rather than "reciprocal".

By the end of this chapter, you should be able to:

- recognize which further rules of algebra do and do not apply to matrix multiplication;
- correctly use inverses, recognizing that not all nonzero matrices are invertible;
- determine when a 2×2 matrix has an inverse, and compute the inverse when it exists;
- use 2×2 inverses to run one step of Newton's method to find zeros of non-linear functions.

18.1. Inverse of a matrix. A function takes in an input and produces an output. Some functions have unambiguous *inverses* that allow you to go backwards from the output to the input.

Example 18.1.1. The function $f(x) = 2x + 3$ from $\mathbf{R}$ to $\mathbf{R}$ has an unambiguous inverse because given $y = 2x + 3$ we can "solve for x in terms of y": $x = (y - 3)/2$. We say the function $g(y) = (y - 3)/2$ is the *inverse function* to f because

$$g(f(x)) = x \text{ for all } x, \quad \text{and} \quad f(g(y)) = y \text{ for all } y.$$

(Explicitly, $g(f(x)) = (f(x)-3)/2 = ((2x+3)-3)/2 = x$ and $f(g(y)) = 2g(y)+3 = 2((y-3)/2)+3 = y$.) The feature that the compositions $g \circ f$ and $f \circ g$ *both* "don't do anything" to x and y, respectively, is the key point. ■

Example 18.1.2. The function $f : \mathbf{R} \to \mathbf{R}$ defined by $f(x) = x^2$ does not have an unambiguous inverse. An illustration of the problem is that the inputs $\sqrt{2}$ and $-\sqrt{2}$ *both* produce the output 2. So, while we often think of $g(y) = \sqrt{y}$ for $y \geq 0$ as a way of reversing the function $f(x) = x^2$, at least when $x \geq 0$, it is not unambiguous since we can also use $-\sqrt{y}$. There are additional problems when $y < 0$. ■

Example 18.1.3. The function $f : \mathbf{R}^2 \to \mathbf{R}^2$ defined by $f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 2x_1 + x_2 \\ x_1 + 2x_2 \end{bmatrix}$ has an unambiguous inverse: writing $f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$, so $2x_1 + x_2 = y_1$ and $x_1 + 2x_2 = y_2$, we can "solve for each of x_1 and x_2 in terms of y_1 and y_2". More precisely, by the method for solving a system of 2 equations in 2 unknowns taught in algebra classes, we obtain $x_1 = (2y_1 - y_2)/3$ and $x_2 = (-y_1 + 2y_2)/3$. In the language of vectors $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} (2y_1 - y_2)/3 \\ (-y_1 + 2y_2)/3 \end{bmatrix}$, so an *inverse function* to f is $g\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = \begin{bmatrix} (2y_1 - y_2)/3 \\ (-y_1 + 2y_2)/3 \end{bmatrix}$. ■

A function with an unambiguous inverse is called *invertible* because we can unambiguously "undo" its effect. Let's recast Example 18.1.3 in the language of matrices, to see it as part of a wider context.

Example 18.1.4. Consider the 2×2 matrix $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$. The associated linear transformation $T_A : \mathbf{R}^2 \to \mathbf{R}^2$ that assigns to each input vector $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbf{R}^2$ the output vector $A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbf{R}^2$ is given by the recipe

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2x_1 + x_2 \\ x_1 + 2x_2 \end{bmatrix}$$

that is exactly $f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right)$ from Example 18.1.3.

Its inverse function $g : \mathbf{R}^2 \to \mathbf{R}^2$ from Example 18.1.3 *also* has such a description, using another 2×2 matrix:

$$g\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = \begin{bmatrix} (2y_1 - y_2)/3 \\ (-y_1 + 2y_2)/3 \end{bmatrix} = \begin{bmatrix} (2/3)y_1 - (1/3)y_2 \\ (-1/3)y_1 + (2/3)y_2 \end{bmatrix} = \begin{bmatrix} 2/3 & -1/3 \\ -1/3 & 2/3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = T_B\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right)$$

for the 2×2 matrix $B = \begin{bmatrix} 2/3 & -1/3 \\ -1/3 & 2/3 \end{bmatrix}$. ■

In the preceding example, the linear transformation $T_A : \mathbf{R}^2 \to \mathbf{R}^2$ has inverse g that is *also* a linear transformation, written as T_B for some 2×2 matrix B. Can we express the property that T_A and T_B are inverse to each other directly in terms of the matrices A and B? If you compute AB and BA (please do) you'll find that *both* equal the 2×2 identity matrix I_2. This is not a coincidence:

Proposition 18.1.5. For any $n \times n$ matrix A, the following two conditions on A are equivalent:

(a) The linear transformation $T_A : \mathbf{R}^n \to \mathbf{R}^n$ is invertible. Explicitly, for every (output) $\mathbf{b} \in \mathbf{R}^n$ there is a unique (input) $\mathbf{x} \in \mathbf{R}^n$ that solves the equation $A\mathbf{x} = \mathbf{b}$.

(b) There is an $n \times n$ matrix B for which $AB = \mathrm{I}_n$ and $BA = \mathrm{I}_n$ (in which case the function $T_B : \mathbf{R}^n \to \mathbf{R}^n$ is inverse to $T_A : \mathbf{R}^n \to \mathbf{R}^n$), with I_n as in Definition 15.1.4.

When these conditions hold, B is uniquely determined and is denoted A^{-1},

Definition 18.1.6. We call any A satisfying these conditions *invertible*, and then B is called the *inverse matrix* of A (and likewise A is then the inverse matrix of B).

See Remark 18.1.9 for comments on determining invertibility and computing the inverse. Applications of inverses are given in Section 18.3, Section 18.5, Theorem 22.5.4, and Remark 22.5.8.

For those who are interested, see Section 18.6 for a proof of the equivalence of (a) and (b).

Example 18.1.7. For $A = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix}$, consider the function $f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} x_1 + 2x_2 \\ 3x_1 + 5x_2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = T_A\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right)$. For the matrix $B = \begin{bmatrix} -5 & 2 \\ 3 & -1 \end{bmatrix}$, please check directly that AB and BA are each equal to I_2, so we say $\begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix}^{-1} = \begin{bmatrix} -5 & 2 \\ 3 & -1 \end{bmatrix}$.

We can check by direct substitution that the functions T_A and T_B are inverse to each other; i.e., $T_B\left(T_A\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right)\right) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ and $T_A\left(T_B\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right)\right) = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$. For instance, $T_B\left(T_A\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right)\right)$ equals

$$T_B\left(\begin{bmatrix} x_1 + 2x_2 \\ 3x_1 + 5x_2 \end{bmatrix}\right) = \begin{bmatrix} -5 & 2 \\ 3 & -1 \end{bmatrix} \begin{bmatrix} x_1 + 2x_2 \\ 3x_1 + 5x_2 \end{bmatrix} = \begin{bmatrix} -5(x_1 + 2x_2) + 2(3x_1 + 5x_2) \\ 3(x_1 + 2x_2) - (3x_1 + 5x_2) \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

(the final equality involves some cancellation in the algebra that we leave for you to check). Please verify similarly for yourself by direct calculation that $T_A\left(T_B\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right)\right) = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$. ■

Although the definition of invertibility in Definition 18.1.6 treats A and B on equal footing, it is convenient that in practice we only need to check half of the definition:

Theorem 18.1.8. If A and B are $n \times n$ matrices that satisfy $AB = \mathrm{I}_n$ then A is invertible and B is its inverse; i.e., automatically the other equation $BA = \mathrm{I}_n$ holds.

A proof is given in Section B.1 (see Proposition B.1.4(ii)) for those who are interested. The key point is to understand how linear transformations interact with dimensions.

Matrix inverses arise in a wide variety of applications, including computer graphics (for real-time 3-dimensional image rendering involving frequent change of perspective), certain types of wireless communication technology (to recover a message from a transmission signal that is received), and GPS (which we'll return to in Section 18.5).

We did not define what it means for an $m \times n$ matrix to be invertible when $m \neq n$. For a 2×3 matrix A, could there be a 3×2 matrix B for which $AB = \mathrm{I}_2$ and $BA = \mathrm{I}_3$? If you think about it geometrically via linear transformations (as rotations, shearings of space, projections, or compositions thereof), it is hard to imagine how a linear transformation $T_A : \mathbf{R}^3 \to \mathbf{R}^2$ that squashes all of 3-dimensional space (linearly) into a plane could be inverted. This is a nice instance of using the visualization of linear transformations to gain insight into what is otherwise pure algebra with matrices. The non-invertibility of any 2×3 matrix is visually suggested by Figure 18.1.1, and Proposition B.1.4(i) shows that this intuition is correct (and that more generally there is no sensible inverse for any $m \times n$ matrix whenever $m \neq n$).

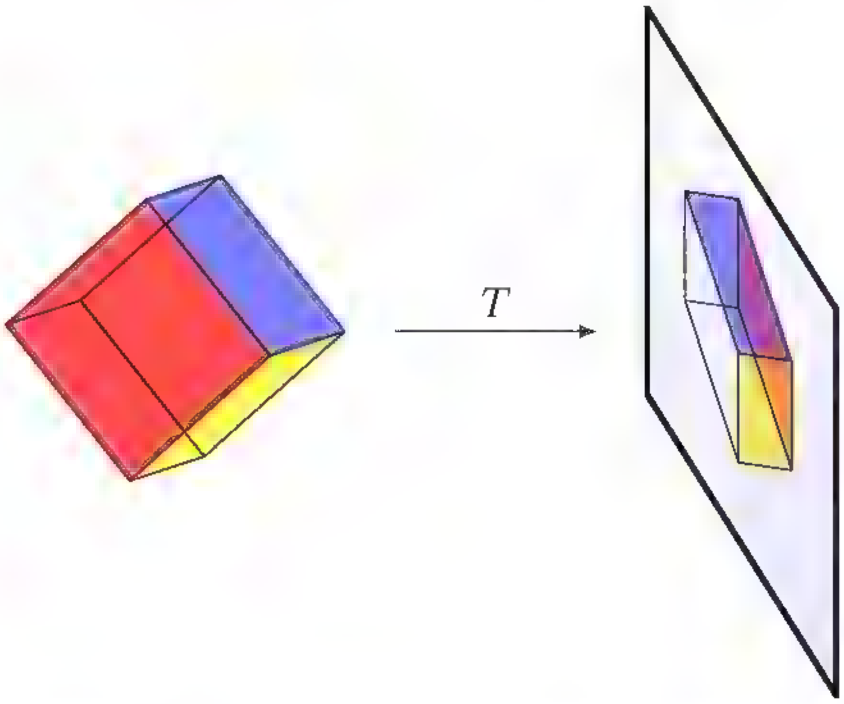


FIGURE 18.1.1. The image of a solid cube (with white unseen faces) under a linear transformation $T : \mathbf{R}^3 \to \mathbf{R}^2$. Many points of the *solid* cube go onto the same point in $\mathbf{R}^2$; any point in $\mathbf{R}^2$ hit by two cube faces with different colors is colored by the hybrid color.

Remark 18.1.9. It is *very important* to be able to determine if an $n \times n$ matrix A is invertible, and to compute A^{-1} when it exists. There is an explicit formula for inverses: for $n = 2$ it will be given in (18.2.2), but for $n > 2$ the formula is very complicated and also impractical to use (even by computers). See Remark E.4.11 if you are curious.

There is one "easy" case: diagonal matrices! Just as multiplication for $n \times n$ diagonal matrices is multiplying corresponding entries (Proposition 14.3.7), for inverses we just reciprocate the entries:

$$\text{if } A = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_n \end{bmatrix} \text{ with all } a_i \neq 0 \text{ then } A^{-1} = \begin{bmatrix} 1/a_1 & 0 & \cdots & 0 \\ 0 & 1/a_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1/a_n \end{bmatrix} \tag{18.1.1}$$

(and if any diagonal entry of a *diagonal* matrix vanishes then it is *not* invertible).

Efficient software works well for matrices with explicit numbers as entries when $n > 2$ is not too huge; that is often the best course of action for practical problems. However, certain types of $n \times n$ matrices are easy to invert (see Section 22.3 for the upper triangular case), and this can be bootstrapped to understand more general cases. It is a challenging and interesting problem to find efficient algorithms to invert $n \times n$ matrices. We will return to this in Chapter 22.

A "random" $n \times n$ matrix is invertible (we explain the precise meaning of this in Remark E.4.6 and Theorem E.4.10), so an $n \times n$ matrix that is non-invertible is rather special. An archaic synonym for "special" is "singular", so it is common in many references to call a non-invertible $n \times n$ matrix *singular*, and so to call invertible $n \times n$ matrices *non-singular*.

18.2. The 2×2 case and determinants.

Example 18.2.1. A 2×2 matrix $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ has an inverse if $ad - bc \neq 0$. To understand this, for the system of linear equations

$$\begin{aligned} ax_1 + bx_2 &= y_1 \\ cx_1 + dx_2 &= y_2 \end{aligned}$$

(which in vector language says $A\mathbf{x} = \mathbf{y}$) let's try to "solve for the x's in terms of the y's". We get rid of x_2 by multiplying the first equation by d, the second by b (so we have bdx_2 in each), and subtracting the second from the first: this yields $(ad - bc)x_1 = dy_1 - by_2$, so dividing by the nonzero $ad - bc$ solves for x_1. Similarly we get rid of x_1 by multiplying the first equation by c, the second by a (so we have acx_1 in each), and subtracting the second from the first: this yields $(bc - ad)x_2 = cy_1 - ay_2$, so dividing by the nonzero $bc - ad = -(ad - bc)$ solves for x_2. Putting it all together, we get

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} (ad-bc)^{-1}(dy_1 - by_2) \\ (bc-ad)^{-1}(cy_1 - ay_2) \end{bmatrix} = \frac{1}{ad-bc}\begin{bmatrix} dy_1 - by_2 \\ -cy_1 + ay_2 \end{bmatrix} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}. \tag{18.2.1}$$

In other words, the matrix

$$\frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \begin{bmatrix} d/(ad-bc) & -b/(ad-bc) \\ -c/(ad-bc) & a/(ad-bc) \end{bmatrix} \tag{18.2.2}$$

is the inverse A^{-1}, because it expresses $\mathbf{x}$ in terms of $\mathbf{y}$ as in (18.2.1). (You can also check directly that the product of the right side of (18.2.2) against A in both orders is equal to I_2.) The formula (18.2.2) is *very useful* in practice, and here is a way to remember how to build it from A: swap the diagonal entries, multiply the off-diagonal entries by -1, and divide by the "cross-difference" $ad - bc$.

To illustrate (18.2.2) numerically, consider $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ as in Example 18.1.3, or $A = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix}$ as in Example 18.1.7. The formula (18.2.2) in these two cases replicates the matrix B in each of those examples. (Check this by plugging the values of a, b, c, d from each of these cases into (18.2.2).) ■

Example 18.2.2. We saw in Section 14.4 that the rotation counterclockwise in $\mathbf{R}^2$ by an angle θ (in radians) has the matrix

$$A_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

Does this have an inverse? One approach is to use the criterion as in Example 18.2.1 upon noting that in this case $ad - bc = (\cos\theta)(\cos\theta) - (-\sin\theta)(\sin\theta) = \cos^2\theta + \sin^2\theta = 1 \neq 0$.

It is also illuminating to think about what is going on geometrically rather than algebraically: can we undo the effect of a counterclockwise rotation by the given angle θ? Indeed we can, simply rotating by the same angle in the opposite direction (i.e., clockwise), which has the same effect as rotating by $2\pi - \theta$

counterclockwise. For instance, in terms of degrees, the effect of a $-50°$ rotation counterclockwise is the same as both a $310°$ rotation counterclockwise and a $50°$ rotation *clockwise*, as in Figure 18.2.1.

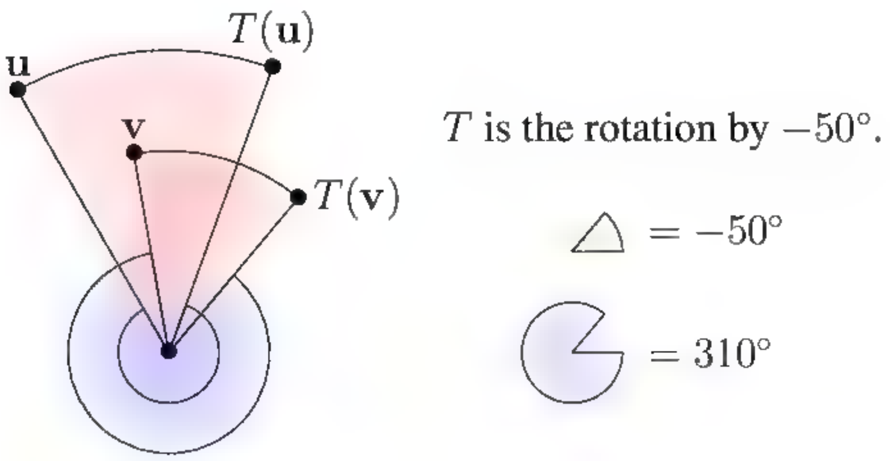


FIGURE 18.2.1. The effect of rotating 310° one way is the same as rotating $-50°$ the same way as well as 50° the other way.

In symbols, $A_\theta^{-1} = A_{2\pi-\theta} = A_{-\theta}$, the final equality holding since adding 2π to an angle (in radians) has no effect on the associated rotation.

You should also check $A_{-\theta}$ is inverse to A_θ directly via matrix multiplication: compute that $A_\theta A_{-\theta}$ and $A_{-\theta}A_\theta$ both equal I_2 (using that $\sin(-\theta) = -\sin(\theta)$, $\cos(-\theta) = \cos(\theta)$, and $\sin(\theta)^2 + \cos(\theta)^2 = 1$). ■

Remark 18.2.3. It is also true that if $ad - bc = 0$ then the matrix $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is *not* invertible. The reason is essentially embedded in the formula (18.2.2): if an inverse were to exist then it would have to be given by this expression, which does not make sense in this case.

For those who are interested, here is a way to show that A really *cannot* ever admit an inverse when $ad - bc = 0$. We will show that the equation $A\mathbf{x} = \mathbf{0}$ always has at least two solutions for such A, so A cannot have an inverse (since by Definition 18.1.6 invertibility is equivalent to the equation $A\mathbf{x} = \mathbf{b}$ having *exactly one* solution for *every* $\mathbf{b} \in \mathbf{R}^2$; we'll violate this with $\mathbf{b} = \mathbf{0}$). If A is the zero matrix then $A\mathbf{x} = \mathbf{0}$ has lots of solutions: every 2-vector $\mathbf{x}$ works! Next, suppose $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is not the zero matrix, so at least one of the entries a, b, c, d is nonzero. Hence, *at least one* (possibly both) of the vectors $\begin{bmatrix} d \\ -c \end{bmatrix}$ and $\begin{bmatrix} b \\ -a \end{bmatrix}$ is different from $\mathbf{0}$. But direct calculation shows that each of these vectors is a solution to $A\mathbf{x} = \mathbf{0}$ precisely because $ad - bc = 0$:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} d \\ -c \end{bmatrix} = \begin{bmatrix} ad - bc \\ cd - dc \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} b \\ -a \end{bmatrix} = \begin{bmatrix} ab - ba \\ cb - da \end{bmatrix} = \begin{bmatrix} 0 \\ -(ad - bc) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

This exhibits at least one *nonzero* solution to $A\mathbf{x} = \mathbf{0}$, yet $\mathbf{x} = \mathbf{0}$ is also a solution, so $A\mathbf{x} = \mathbf{0}$ has at least two solutions (namely, $\mathbf{0}$ and some nonzero 2-vector), so A is not invertible.

Definition 18.2.4. Let $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$. The *determinant* of A, denoted as $\det(A)$, is $ad - bc$.

Example 18.2.1 and Remark 18.2.3 say that a 2×2 matrix A is invertible precisely when $\det(A) \neq 0$. Thus, to check if a 2×2 matrix A is invertible we compute if $\det(A)$ is nonzero.

Remark 18.2.5. There is a concept of "determinant" $\det(A)$ (a certain scalar) for $n \times n$ matrices A for any n. This is developed in Appendix E, where it is shown that the non-vanishing of $\det(A)$ characterizes exactly when A is invertible. The determinant is complicated when $n > 2$, so using it to detect invertibility is not so useful for numerical work when $n > 2$. However, determinants are conceptually important and show up in probability, multivariable integration (where one uses determinants of derivative matrices: see Section E.6), and differential equations. In the case $n = 2$ the determinant will reappear in Theorem 23.3.1. (A method due to Lewis Carroll computes $n \times n$ determinants from many 2×2 cases!)

Remark 18.2.6. The determinant $\det(A)$ of a 2×2 matrix $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ has a geometric interpretation: $|\det(A)|$ is the distortion effect on area of regions R in $\mathbf{R}^2$ upon applying the associated linear transformation $T_A : \mathbf{R}^2 \to \mathbf{R}^2$ to the points of such a region. That is, if we denote by $T_A(R)$ the collection of points $T_A(\mathbf{x}) = A\mathbf{x}$ for $\mathbf{x} \in R$ then $T_A(R)$ has area $|\det(A)|\,\mathrm{area}(R)$. We explain this here for a special case that is rather illuminating. (For an extension to $n \times n$ matrices, see Section E.3 in Appendix E.)

Consider the unit square $S = \{(x, y) \in \mathbf{R}^2 : 0 \leq x, y \leq 1\}$ with area 1, shown on the left in Figure 18.2.2. Taking this to be the region R, the claim is that $T_A(S)$ has area $|\det(A)| = |ad - bc|$. Swap the columns of A (negating $\det(A)$, hence the need for the absolute value) if necessary to ensure the second column is counterclockwise from the first with angle from the first to the second at most 180°. Figure 18.2.2 illustrates that $T_A(S)$ a parallelogram, showing when the angle between the columns is obtuse.

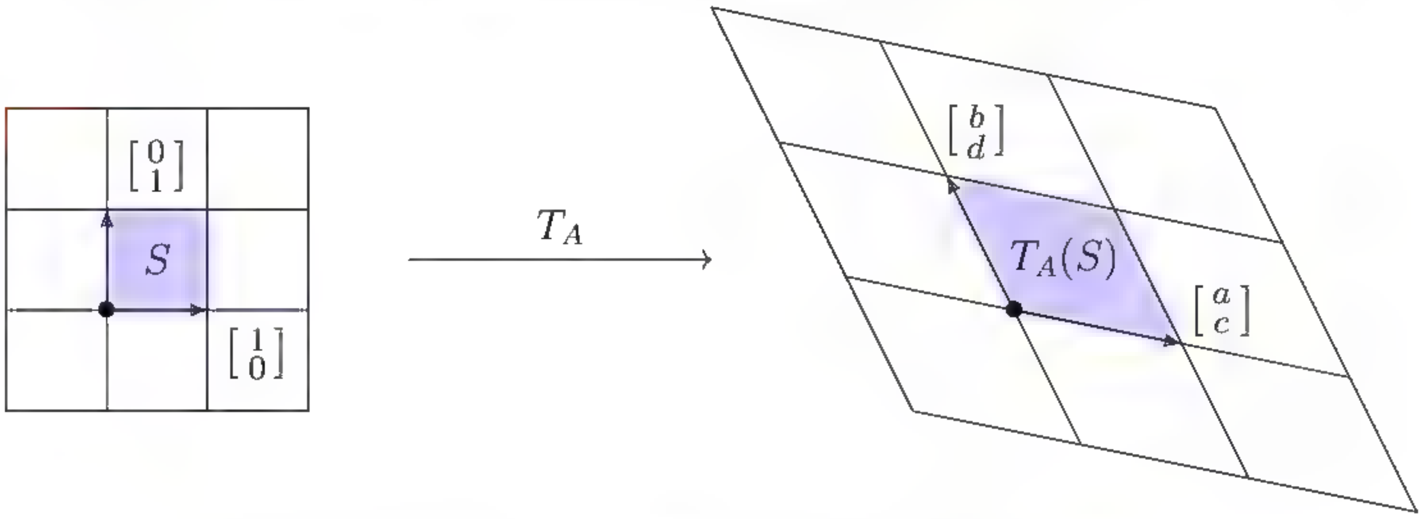

$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$
S
$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$
T_A
$\begin{bmatrix} b \\ d \end{bmatrix}$
$T_A(S)$
$\begin{bmatrix} a \\ c \end{bmatrix}$

FIGURE 18.2.2. The effect of T_A on the unit square S.

The parallelogram $P = T_A(S)$ in Figure 18.2.2 lies inside a bigger rectangle (with sides parallel to the coordinate axes) as in Figure 18.2.3 that we shall decompose as shown to obtain the area of P by subtracting off areas of various triangles and smaller rectangles from the area of the ambient big rectangle.

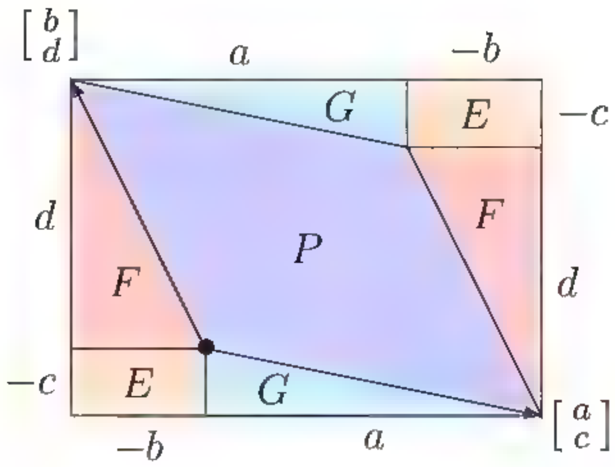

$\begin{bmatrix} b \\ d \end{bmatrix}$
a
$-b$
G
E
$-c$
d
F
P
F
d
$-c$
E
G
$\begin{bmatrix} a \\ c \end{bmatrix}$
$-b$
a

FIGURE 18.2.3. The geometry underlying why the area of $P = T_A(S)$ equals $|\det(A)|$.

The visualization in Figure 18.2.3 expresses the case that the interior angle of P at the origin is obtuse. We will focus on treating this when (as in Figure 18.2.3) the bottom edge dips below the x-axis and the left edge lies to the left of the y-axis; algebraically this says $c < 0$, $a > 0$, $b < 0$, and $d > 0$. (The other possibilities with an obtuse interior angle at the origin can be reduced to this case via arguments with rotations that we omit; cases with interior angle $\leq 90^\circ$ at the origin are handled similarly.) This yields

$$\text{area}(E) = (-b)(-c) = bc, \quad \text{area}(F) = \frac{1}{2}(-b)d = -\frac{bd}{2}, \quad \text{area}(G) = \frac{1}{2}(-c)a = -\frac{ac}{2}$$

(keep in mind that $-b, -c > 0$). Now we can compute the (positive) area of P via some algebra:

$$\begin{aligned} \text{area(large rectangle)} - 2\,\text{area}(E) - 2\,\text{area}(F) - 2\,\text{area}(G) &= (a-b)(d-c) - 2bc + bd + ac \\ &= (ad + bc - ac - bd) - 2bc + bd + ac \\ &= ad - bc \\ &= \det(A). \end{aligned}$$

Remark 18.2.7 (online resource). The first half of this video at "Essence of Linear Algebra" vividly illustrates the meaning of determinants for 2 × 2 matrices in the spirit of the preceding discussion.

Example 18.2.8. An elegant synthesis of algebraic and geometric perspectives on linearity is the "pure thought" determination of the average area of the shadow of a cube, explained here. This also illustrates the power of generalization for solving specific problems in mathematics. ■

18.3. Application of matrix inverses I: linear systems. One of the most significant applications of matrix inverses is to provide a systematic way to think about the process of solving systems of linear equations for which the number of equations is the *same* as the number of unknowns:

Example 18.3.1. Let's solve the system of 3 linear equations in 3 unknowns, visualized in Figure 18.3.1:

$$\begin{cases} 2x + y - z = 4 \\ x - 2y + z = 3 \\ x - y - z = 1 \end{cases} . \tag{18.3.1}$$

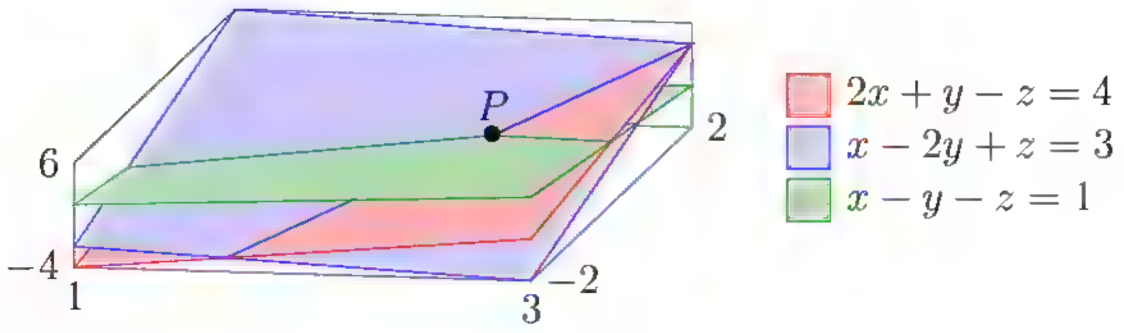


FIGURE 18.3.1. Three planes, crossing pairwise in lines that all meet at a common point P

We will solve this system of equations using a matrix inverse. First, rewrite the system in matrix form:

$$\underbrace{\begin{bmatrix} 2 & 1 & -1 \\ 1 & -2 & 1 \\ 1 & -1 & -1 \end{bmatrix}}_{A} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix} . \tag{18.3.2}$$

Your computer tells you the good news that A is invertible, with

$$A^{-1} = \frac{1}{7}\begin{bmatrix} 3 & 2 & -1 \\ 2 & -1 & -3 \\ 1 & 3 & -5 \end{bmatrix} = \begin{bmatrix} 3/7 & 2/7 & -1/7 \\ 2/7 & -1/7 & -3/7 \\ 1/7 & 3/7 & -5/7 \end{bmatrix}.$$

Now comes the great idea: if we multiply both sides of (18.3.2) by A^{-1} we get $A^{-1}A\begin{bmatrix} x \\ y \\ z \end{bmatrix} = A^{-1}\begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix}$, or in numbers,

$$\begin{bmatrix} 3/7 & 2/7 & -1/7 \\ 2/7 & -1/7 & -3/7 \\ 1/7 & 3/7 & -5/7 \end{bmatrix}\begin{bmatrix} 2 & 1 & -1 \\ 1 & -2 & 1 \\ 1 & -1 & -1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3/7 & 2/7 & -1/7 \\ 2/7 & -1/7 & -3/7 \\ 1/7 & 3/7 & -5/7 \end{bmatrix}\begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix}.$$

That may look like a disaster, but it is not: the point is that the product of the two 3×3 matrices on the left side collapses to I_3 precisely because it is $A^{-1}A$. Hence, the left side simplifies dramatically: the contribution from the 3×3 matrices disappears, leaving us with the equation

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3/7 & 2/7 & -1/7 \\ 2/7 & -1/7 & -3/7 \\ 1/7 & 3/7 & -5/7 \end{bmatrix}\begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 17/7 \\ 2/7 \\ 8/7 \end{bmatrix}.$$

Thus, the original linear system (18.3.1) has as its unique solution $x = 17/7$, $y = 2/7$, $z = 8/7$.

We can write this all more succinctly in the following way:

$$A\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix} \text{ implies } \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \left(A^{-1}A\right)\begin{bmatrix} x \\ y \\ z \end{bmatrix} = A^{-1}\left(A\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = A^{-1}\begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 17/7 \\ 2/7 \\ 8/7 \end{bmatrix}.$$

We are using here the fact that we can regroup the factors in the product $A^{-1}(A\mathbf{v}) = (A^{-1}A)\mathbf{v}$, which we discussed in Chapter 15.3. (This also works in reverse: if $\mathbf{v} = A^{-1}\mathbf{b}$ then $A\mathbf{v} = A(A^{-1}\mathbf{b}) = (AA^{-1})\mathbf{b} = \mathrm{I}_3\mathbf{b} = \mathbf{b}$.)

The method of multiplying by A^{-1} on the left works for *any* values put on the right side of (18.3.1) to solve for (x, y, z). For example, if instead we had different constants on the right side, such as

$$\begin{cases} 2x + y - z = -2 \\ x - 2y + z = -1 \\ x - y - z = 5 \end{cases}$$

(but the left side is *the same* as (18.3.1)) then we again rewrite the system in matrix form as

$$\underbrace{\begin{bmatrix} 2 & 1 & -1 \\ 1 & -2 & 1 \\ 1 & -1 & -1 \end{bmatrix}}_{\text{same } A!}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -2 \\ -1 \\ 5 \end{bmatrix}$$

and multiply by the *same* A^{-1} on the left to obtain

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3/7 & 2/7 & -1/7 \\ 2/7 & -1/7 & -3/7 \\ 1/7 & 3/7 & -5/7 \end{bmatrix}\begin{bmatrix} -2 \\ -1 \\ 5 \end{bmatrix} = \begin{bmatrix} -13/7 \\ -18/7 \\ -30/7 \end{bmatrix}.$$

■

Unfortunately the method in Example 18.3.1 is *not* completely general: it does not work for every possible system of n linear equations in n unknowns. The reason is that the $n \times n$ matrix associated with this system of equations may *not* be invertible. In such cases this method fails, and it requires more insight to understand whether there are solutions to such a "singular" system.

Let's illustrate the effect of non-invertibility with an example:

Example 18.3.2. Consider the system of equations

$$\begin{cases} x + 3y - z = 4 \\ 2x + y + 3z = 3 \\ -x + y - 3z = 1 \end{cases},$$

or equivalently, in matrix form,

$$\underbrace{\begin{bmatrix} 1 & 3 & -1 \\ 2 & 1 & 3 \\ -1 & 1 & -3 \end{bmatrix}}_{A} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix}. \tag{18.3.3}$$

Though it is probably not apparent based on what we have discussed so far, this matrix A is *not* invertible and moreover the system of equations turns out *not* to have any solution at all. In Chapter 21 we will explain how to obtain such conclusions in a systematic way.

We should be clear: non-invertibility of the matrix A of coefficients does not always prevent the existence of solutions! For example, if we look at a system of equations which is slightly different from (18.3.3), where the left side (the matrix, or equivalently, the coefficients of x, y and z) remains the same but we alter the constants on the right side, then different things can happen. As an illustration, replace $\begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix}$ by $\begin{bmatrix} 5 \\ 5 \\ -1 \end{bmatrix}$ on the right side; i.e., consider the modified system of equations

$$\underbrace{\begin{bmatrix} 1 & 3 & -1 \\ 2 & 1 & 3 \\ -1 & 1 & -3 \end{bmatrix}}_{\text{same } A!} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \\ -1 \end{bmatrix}.$$

This system turns out to have a solution, such as $(x, y, z) = (2, 1, 0)$, and it even has *lots more* solutions, such as $(-2, 3, 2)$, $(6, -1, -2)$. In fact, *every* point on the parametric line $\begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} + t \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix}$ (for any $t \in \mathbf{R}$) is a solution! (We will explain how to understand this systematically in Chapters 21 and 22.) ■

The preceding examples illustrate the following general result.

Theorem 18.3.3. Let A be an $n \times n$ matrix, and consider the system of n equations in n unknowns $A\mathbf{x} = \mathbf{b}$, where $\mathbf{b}$ is a given n-vector.

(i) If A is invertible then this system has a unique solution, namely $\mathbf{x} = A^{-1}\mathbf{b}$. In particular (taking $\mathbf{b} = \mathbf{0}$), $A\mathbf{x} = \mathbf{0}$ has $\mathbf{x} = \mathbf{0}$ as its *only* solution when A is invertible.

(ii) if A is *not* invertible then, depending on which n-vector $\mathbf{b}$ we choose, the system $A\mathbf{x} = \mathbf{b}$ either has no solution or has infinitely many solutions. In particular (taking $\mathbf{b} = \mathbf{0}$, so

$A\mathbf{x} = \mathbf{b}$ does have a solution, namely $\mathbf{x} = \mathbf{0}$), when A is not invertible the vector equation $A\mathbf{x} = \mathbf{0}$ has nonzero solutions (in fact, infinitely many such).

Thus, A is invertible precisely when $A\mathbf{x} = \mathbf{0}$ has $\mathbf{x} = \mathbf{0}$ as its only solution, and A is non-invertible precisely when $A\mathbf{x} = \mathbf{0}$ has a nonzero solution.

The invertible case was illustrated in Example 18.3.1, and the two possible outcomes in the non-invertible case were illustrated in Example 18.3.2. It may be surprising that the two outcomes in (ii) when A is non-invertible (no solutions or infinitely many solutions) are the *only* possibilities. For example, if A is a non-invertible $n \times n$ matrix then it is impossible that $A\mathbf{x} = \mathbf{b}$ has precisely five solutions. This is explained in Section B.2, building on Chapter 21 (Corollary B.2.8(b) addresses the essential step: showing an $n \times n$ matrix A is invertible when $A\mathbf{x} = \mathbf{0}$ has $\mathbf{x} = \mathbf{0}$ as the only solution).

This discussion leaves us some issues unresolved: how do we determine if a given $n \times n$ matrix A is invertible (equivalently: does $A\mathbf{x} = \mathbf{0}$ have $\mathbf{x} = \mathbf{0}$ as the only solution)? If it is invertible, how do we compute A^{-1}? When A is not invertible, how can we understand for a given $\mathbf{b}$ if $A\mathbf{x} = \mathbf{b}$ has no solutions or infinitely many? We address these matters in Chapters 21–22.

18.4. More properties of matrix algebra. Recall the following from Section 15.3:

$A(B+C) = AB + AC$, $(A+B)C = AC + BC$, and $A(BC) = (AB)C$, but $AB \neq BA$ **in general**!

We add to these some further properties of matrix multiplication which involve inverses of matrices.

(i) **When** A **is invertible** you can "cancel A" by multiplying both sides by A^{-1} (but there is a caveat; see the Warning below):
  - Cancelling an invertible matrix on the left: if $AB = AC$ **and** A **is invertible** then $B = C$. This holds because you multiply both sides *on the left* by A^{-1}.
  - Cancelling an invertible matrix on the right: if $BA = CA$ **and** A **is invertible** then $B = C$. For this you need to multiply both sides *on the right* by A^{-1}.
  - **Warning**: our caveat is that if $AB = CA$, then you cannot cancel A on the left on one side and on the right on the other, so you *cannot conclude* in this case that $B = C$, even when A is invertible (see Example 18.4.1 below).

(ii) If A and B are both invertible $n \times n$ matrices then AB is also invertible, and $(AB)^{-1} = B^{-1}A^{-1}$ (**note the switch of order of multiplication on the right side!**); see Example 18.4.3 below for an illustration.

Example 18.4.1. Here is an example where $AB = CA$ and A is invertible, but $B \neq C$. The matrix $A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}$ is invertible, with inverse $A^{-1} = \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix}$ (you can check directly that the product of these matrices is I_2, or you can use the procedure in Example 18.2.1). However, for the visibly distinct matrices

$$B = \begin{bmatrix} 2 & 0 \\ 0 & -3 \end{bmatrix}, \quad C = \begin{bmatrix} -8 & 5 \\ -10 & 7 \end{bmatrix},$$

you can check by multiplying matrices that

$$AB = \begin{bmatrix} 2 & -3 \\ 4 & -3 \end{bmatrix} = CA.$$

More generally, for invertible $n \times n$ matrices A with $n > 1$, if $AB = CA$ then it is almost always the case that $B \neq C$. ■

Remark 18.4.2. There are important contexts where it is useful to consider matrices A, B, and C, with A invertible, B diagonal, but C non-diagonal (so $C \neq B$) for which $AB = CA$. This relates to the fundamental notion of *eigenvectors* of a square matrix that we introduce in Chapter 23; this notion is extremely relevant to data science, image compression, dynamical systems, quantum mechanics, and much more.

Example 18.4.3. Here is an example which illustrates the general fact that $(AB)^{-1} = B^{-1}A^{-1}$ when A and B are invertible, though typically $(AB)^{-1} \neq A^{-1}B^{-1}$ (i.e., the order of multiplication of A^{-1} and B^{-1} **really matters**).

Consider the same A and B as in Example 18.4.1:

$$A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}, \quad A^{-1} = \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 0 \\ 0 & -3 \end{bmatrix}, \quad B^{-1} = \begin{bmatrix} 1/2 & 0 \\ 0 & -1/3 \end{bmatrix}.$$

(The determination of B^{-1} is easy because B is diagonal with nonzero diagonal entries: see (18.1.1).) You can check that $AB = \begin{bmatrix} 2 & -3 \\ 4 & -3 \end{bmatrix}$, as in Example 18.4.1, and also that

$$(AB)^{-1} = \begin{bmatrix} -1/2 & 1/2 \\ -2/3 & 1/3 \end{bmatrix}.$$

(It is easy to check directly that the product of the right side with AB equals I_2, or alternatively, you can use the general formula from Example 18.2.1). Multiplying the matrices directly shows that

$$B^{-1}A^{-1} = \begin{bmatrix} 1/2 & 0 \\ 0 & -1/3 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix} = \begin{bmatrix} -1/2 & 1/2 \\ -2/3 & 1/3 \end{bmatrix},$$

which is exactly the same as $(AB)^{-1}$. However, multiplying these inverses in the other order gives a (very) *different* matrix:

$$A^{-1}B^{-1} = \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 1/2 & 0 \\ 0 & -1/3 \end{bmatrix} = \begin{bmatrix} -1/2 & -1/3 \\ 1 & 1/3 \end{bmatrix}. \qquad \blacksquare$$

Example 18.4.4. As an alternative to the algebraic approach in Example 18.4.3, there is a nice geometric way to think about the formula $(AB)^{-1} = B^{-1}A^{-1}$. This explains the reason for the order in which we multiply A^{-1} and B^{-1}.

Consider two invertible linear transformations T and T' on $\mathbf{R}^n$, such as rotations of $\mathbf{R}^3$ by some specific respective angles θ and θ' around specific respective lines ℓ and ℓ' passing through the origin. Then $T' \circ T$ encodes the overall effect of *first* applying the transformation T and *then* applying T'. As an explicit example, if T is the rotation R_z around the z-axis corresponding to a 90-degree counterclockwise rotation in the xy-plane and T' is the rotation R_x around the x-axis corresponding to a 90-degree counterclockwise rotation in the yz-plane then $T' \circ T = R_x \circ R_z$ corresponds to the product matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}.$$

This final matrix may look mysterious, but it turns out to be a rotation (of 120 degrees counterclockwise) around the line through $(1, -1, 1)$ and $\mathbf{0}$. There is no reason you would recognize that immediately without more experience. However, as a safety check you can verify that this final matrix does carry the point $(1, -1, 1) \in \mathbf{R}^3$ to itself (which is the same as what a rotation around the line through that point and $\mathbf{0}$ does), and we give an illustration in Figure 18.4.1 (follow the red dot in one corner).

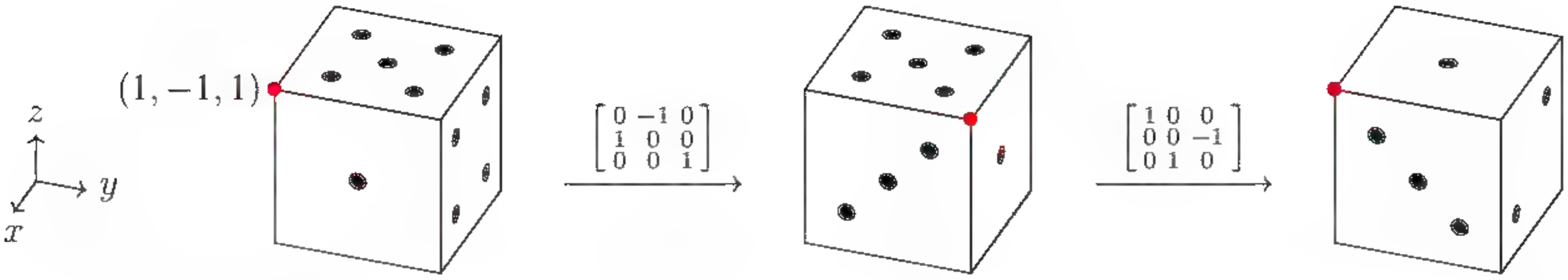

z
y
x
$(1,-1,1)$
$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$

FIGURE 18.4.1. The effect of R_z and R_x on a cube. The net transformation of the cube is $R_x \circ R_z$.

How can we *undo* the composite operation $T' \circ T$? In terms of the example with rotations in $\mathbf{R}^3$, to undo $R_x \circ R_z$ we undo the steps in the *opposite* order from which they were applied. (Why is that reasonable? Think about undoing other invertible processes, such as: putting on socks and shoes, or opening a window and sticking your head out. During this lecture in Fall 2018, a student removed a sock while keeping his shoes on but said it was very painful.) Thus, we should first undo R_x and then undo R_z. Likewise in general we should *first* undo the last step, namely undo T', and *then* undo the first step, namely undo T.

Undoing an invertible linear transformation corresponds to applying the inverse transformation (encoded by the inverse matrix), so we conclude that the effect of undoing $T' \circ T$ is given by $T^{-1} \circ T'^{-1}$, which is to say

$$(T' \circ T)^{-1} = T^{-1} \circ T'^{-1}.$$

Thus we have expressed in the language of linear transformations the formula $(AB)^{-1} = B^{-1}A^{-1}$ for invertible matrices. This explains why the order of multiplication of A^{-1} and B^{-1} is what it is. ■

18.5. Application of matrix inverses II: Newton's method for finding zeros. An important application where inverses of matrices appear in a crucial way is *Newton's method*, an algorithm that numerically approximates solutions to a vector equation $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, where $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^n$ is a *non-linear* function. (When $\mathbf{f}$ is a *linear* function, finding solutions to $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ is an important task in linear algebra; we will discuss it in Chapters 21–22.) Explicitly, if $f_j : \mathbf{R}^n \to \mathbf{R}$ is the jth component function of $\mathbf{f}$ – i.e., $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_n(\mathbf{x}))$ – then $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ expresses a system of n *non-linear* equations in n unknowns:

$$f_1(x_1, \ldots, x_n) = 0, \ldots, f_n(x_1, \ldots, x_n) = 0.$$

Before discussing what Newton's method actually is, let us give some specific instances with various functions $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^n$; we will then revisit these examples after we have formulated Newton's method.

Example 18.5.1. Suppose we want to find a simultaneous solution to the pair of (non-linear) equations

$$x^3 + xy + 2y^3 - 3 = 0, \;\; x^2y + xy^2 - y = 0.$$

This is equivalent to trying to solve $\mathbf{f}(x, y) = (0, 0)$, where the function

$$\mathbf{f}(x, y) = (x^3 + xy + 2y^3 - 3, x^2y + xy^2 - y)$$

was already considered in Example 13.5.5. (We computed $(D\mathbf{f})(\mathbf{x})$ there.) You can already imagine the difficulties of trying to find a solution explicitly or by some sort of elementary expression. It looks pretty hopeless, and almost certainly is. This leads one to try to find ways to approximate a solution (or solutions) to this equation as accurately as desired. Already in the 1660's Newton realized the importance of this and invented a method, at least for functions of one variable (see Remark 18.5.5). It turned out to be an extremely efficient method and generalizations of it are still in use and underlie a lot of modern scientific computation, such as numerically estimating solutions to differential equations. ■

Example 18.5.2. Let's discuss a fundamental problem in robotics and computer animation (for 3D images in video games, movies, etc.) where rapidly solving non-linear equations (via Newton's method) plays an essential role. Consider an industrial robot consisting of several mechanically-linked "arms" that are connected one after the other. Each arm can swivel at its link with the previous arm. Suppose that the joint linking each arm to the previous one is constrained to lie in some fixed plane relative to the previous arm. This plane changes from joint to joint, but we only need to specify a single angle at each joint to describe how successive arms lie relative to one another.

Knowing the angle $\theta_1, \theta_2, \dots$ by which each arm is positioned relative to the previous arm determines the entire state of the mechanical system. In particular, such angles determine the position $\mathbf{p} \in \mathbf{R}^3$ of the tip of the final arm. The formula expressing $\mathbf{p}$ in terms of those angles is *extremely* non-linear, due to the intervention of many trigonometric functions (somewhat like the non-linearity near the end of Example 10.1.1, but much worse); see [AL, (9.13)] for the formula.

A similar situation arises in computer graphics (for video games, Pixar, etc.): a computer image of a moving figure has an underlying "skeleton" consisting of rigid parts akin to such a robot. If one prescribes the angle at each joint, the computer image will appear as a specific spatial image. (We are disregarding further spatial information that is useful in practice but not essential for this discussion.)

The key point is that in practice one doesn't want to use the angular data to compute $\mathbf{p}$, or specify the angles at the joints of a computer image to determine the appearance of the image, but rather to go the other way around! Namely, one wants the tip of the last arm of the industrial robot to trace out some specific path $\mathbf{p}(t)$ in space (with t denoting time) and seeks to determine the angles $\theta_1(t), \theta_2(t), \dots$ which allow the robot to continuously achieve that path of motion for its tip. Likewise, a computer graphics artist wants to be able to virtually manipulate the motion of a computer image of a person (or animal, or talking tea cup, etc.) via its position in 3-dimensional space and the software must know how to translate that physical motion into the angle positions that produce it (much as someone throwing a ball thinks in terms of physical positions in space and not about angle measurements).

Underlying all of this, what needs to be done is to solve a non-linear equation $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, or more typically, a (large) sequence of such equations.

Accurately and rapidly calculating the angles from the data of a physical configuration is an entire subject unto itself called "inverse kinematics" which abounds in massively non-linear equations. We discuss this further in Examples 18.5.7 and I.1.4. ■

Example 18.5.3. The Global Positioning System (GPS) is used in daily life by pedestrians, hikers, drivers, airline pilots, etc. navigating via hand-held location devices (such as a smartphone). GPS uses time measurements based on electronic transmissions to determine the distance from your position to each of 4 (moving) satellites, and combines this with the 3-dimensional Pythagorean Theorem (see Theorem 2.3.1) to determine your position (x, y, z) at some time t.

The position (x, y, z) and time t are the *unique* solution to a system of 4 non-linear equations that involve many auxiliary measurements, such as: the positions of the satellites at time t, the time for an electronic signal from each satellite to reach your hand-held device, an error tolerance in the time measurements by your device's internal clock (which is not nearly as accurate as the atomic clocks on the satellites), and the distance from your position to each satellite (computed by a formula involving the speed of the transmission signal through the atmosphere).

There are more than 30 GPS satellites spread across the sky, and typically at least 6 of them are visible to your device. By using input from 4 of the satellites, we want to solve $\mathbf{f}(x, y, z, t) = \mathbf{0}$ for a specific non-linear $\mathbf{f} : \mathbf{R}^4 \to \mathbf{R}^4$; some geometry with spheres ensures that this system of 4 equations in 4 unknowns has a *unique* solution, though the system is very sensitive to seemingly small time measurement errors. (Incorporating more than 4 satellites leads to more equations than unknowns, and new difficulties handled via least-squares techniques.) Your GPS device needs to accurately solve the non-linear system $\mathbf{f}(x, y, z, t) = \mathbf{0}$ *quickly*; GPS also needs to solve other non-linear equations, such as Kepler's equation [KH, Sec. 3.8.1, Table 3.22, (5)]. We will come back to this topic in Example 18.5.8 and discuss it further in Example I.1.4. ■

Example 18.5.4. A very important situation when one wishes to find solutions to a vector equation $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ is when $\mathbf{f}$ is the gradient of a function $g : \mathbf{R}^n \to \mathbf{R}$. Indeed, as we already learned in Chapter 11, to find the critical points – and in particular local minima and maxima – of g we must solve the system of simultaneous equations $\partial g/\partial x_1 = 0, \partial g/\partial x_2 = 0, \ldots, \partial g/\partial x_n = 0$ for the n numbers $x_1, \ldots, x_n$. In vector language, we seek solutions to $(\nabla g)(\mathbf{x}) = \mathbf{0}$.

The method of gradient descent was described in Section 11.3 for locating critical points. However, this can be rather slow, and Newton's method is much faster: it often requires far fewer iterations to produce an approximation with a given degree of accuracy. Novel applications arise in the imaging of black holes, as we alluded to in Example 11.3.4; see Example I.1.3 for more information. ■

Having provided several motivating examples, we now turn to Newton's method itself. When faced with solving a non-linear vector equation $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, we first make some initial guess for a solution. There are many ways that we might choose this, depending on the circumstances and our knowledge about the function, and it obviously is in our best interest not to make a wildly stupid guess. Our goal is to improve an initial guess $\mathbf{a}$ into a better approximation to a solution. We shall do this by finding a small vector $\mathbf{h}$ so that $\mathbf{a} + \mathbf{h}$ is closer to an actual solution than $\mathbf{a}$ is. In practice, one then iterates this: if we call our initial guess $\mathbf{a}_1$, then one application of the algorithm will produce a new and hopefully better approximation $\mathbf{a}_1 + \mathbf{h}$, which we shall call $\mathbf{a}_2$. Repeating this process gives a sequence of points $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \ldots$. The aim is that after computing a moderately small number of such points we should have an approximation to a solution within a high degree of accuracy.

How do we improve our initial guess $\mathbf{a}$ to $\mathbf{a} + \mathbf{h}$ (i.e., what should we choose for $\mathbf{h}$)? The idea is that we think of this as trying to find an $\mathbf{h}$ which solves $\mathbf{f}(\mathbf{a} + \mathbf{h}) = \mathbf{0}$, and solve that by replacing $\mathbf{f}(\mathbf{a} + \mathbf{h})$ by its linear approximation $\mathbf{f}(\mathbf{a}) + (D\mathbf{f})(\mathbf{a})\mathbf{h}$. Of course, "$\mathbf{f}(\mathbf{a}) + (D\mathbf{f})(\mathbf{a})\mathbf{h} = \mathbf{0}$" is not exactly the same equation anymore, but we can hope it is pretty close, and it will be much easier to solve for $\mathbf{h}$!

So let's go ahead and solve this simplified equation. We must *assume* that the $n \times n$ matrix $(D\mathbf{f})(\mathbf{a})$ is invertible. This is key to the method, and if this matrix is not invertible then Newton's method does not apply. (There will also be convergence issues, which are more serious in practice.) We rewrite $\mathbf{f}(\mathbf{a}) + (D\mathbf{f})(\mathbf{a})\mathbf{h} = \mathbf{0}$ as

$$(D\mathbf{f})(\mathbf{a})\mathbf{h} = -\mathbf{f}(\mathbf{a}), \quad \text{and hence} \quad \mathbf{h} = -((D\mathbf{f})(\mathbf{a}))^{-1}\mathbf{f}(\mathbf{a}). \tag{18.5.1}$$

So from our initial guess $\mathbf{a}$ we add on this "correction" $\mathbf{h}$ to produce our next (hopefully more accurate) guess to a solution of the non-linear equation:

$$\mathbf{a} - ((D\mathbf{f})(\mathbf{a}))^{-1}\mathbf{f}(\mathbf{a}).$$

To summarize:

(Newton's method for approximating zeros of non-linear functions) Let $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^n$ be a non-linear function and let $\mathbf{a}$ be an initial guess for a solution to $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, with $(D\mathbf{f})(\mathbf{a})$ invertible. Then if we have chosen $\mathbf{a}$ reasonably, the sequence of vectors $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \ldots \in \mathbf{R}^n$ defined by $\mathbf{a}_1 = \mathbf{a}$ and

$$\mathbf{a}_{k+1} = \mathbf{a}_k - ((D\mathbf{f})(\mathbf{a}_k))^{-1}\mathbf{f}(\mathbf{a}_k) \text{ for all } k \geq 1 \tag{18.5.2}$$

makes sense (e.g., $(D\mathbf{f})(\mathbf{a}_k)$ is invertible for every k) and converges rapidly to a solution of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$.

How do we know whether we have chosen a reasonably? This is of course crucial, and there are lots of examples where this algorithm produces a sequence which does not converge (it may even spiral out to infinity!). An explicit example of this in the setting of Example 18.5.4 with $n = 2$ is given at the end of Example I.2.2. These matters are addressed in courses and books on numerical methods (e.g., [BSt, Sec. 5.3]).

Remark 18.5.5. For $n = 1$, we are trying to approximate a solution to $f(x) = 0$ for a function $f : \mathbf{R} \to \mathbf{R}$. Newton's method says now that if a is our first guess to a solution to $f(x) = 0$ and if $f'(a) \neq 0$, then in favorable circumstances the sequence of numbers $a_1, a_2, a_3, \dots$ defined by $a_1 = a$ and

$$a_{k+1} = a_k - \frac{f(a_k)}{f'(a_k)} \text{ for all } k \geq 1$$

converges rapidly to a solution of $f(x) = 0$. The visualization for this process, in Figure 18.5.1, is as follows. A vertical line (in red) through a_k on the x-axis meets the (light blue) graph $y = f(x)$ at a point, namely $(a_k, f(a_k))$. The tangent line at that point on the graph is $y = f(a_k) + f'(a_k)(x - a_k)$, and this meets the x-axis where $0 = y = f(a_k) + f'(a_k)(x - a_k)$. Solving that (assuming $f'(a_k) \neq 0$!) gives $x = a_k - f(a_k)/f'(a_k) = a_{k+1}$ as the point where the tangent line meets the x-axis, and the process then repeats.

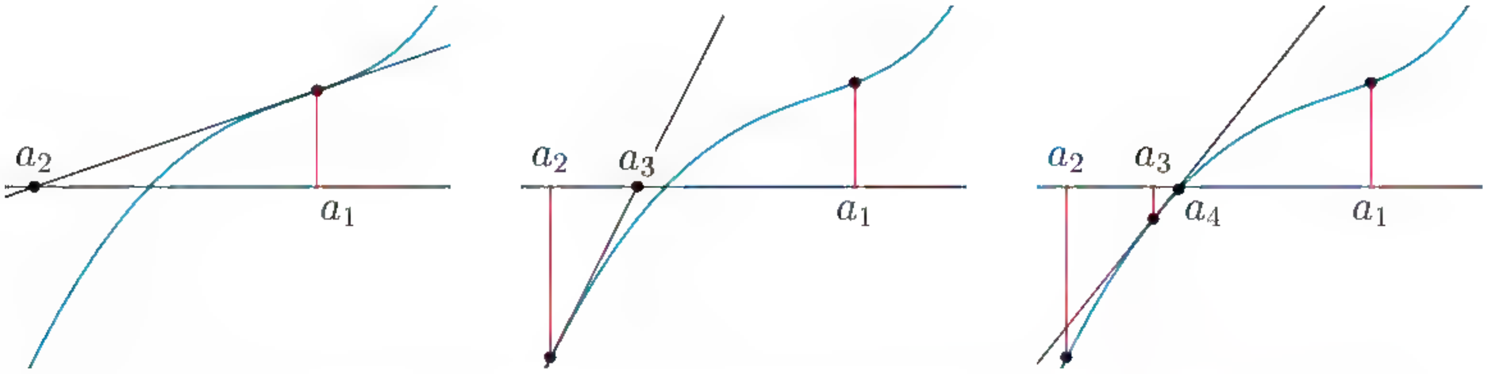


FIGURE 18.5.1. Newton's method demonstrated for the first four points a_1, a_2, a_3, a_4.

Newton's formulation was a cumbersome algebraic process without graphs and derivatives! The modern formulation is due to Simpson [Yp, Sec. 8]. For a visual interpretation of the 2-dimensional case in terms of tangent planes to surface graphs in the spirit of Figure 18.5.1, see this video, where the values of $f_1(x, y)$ and $f_2(x, y)$ at a common point in $\mathbf{R}^2$ are called "residuals" (and see Section 18.7 for an algebraic justification of the visualization in the video).

Even in the 1-dimensional case, Newton's method can fail to converge. Moreover, if $f(x) = 0$ has more than one solution then the method can be *extremely sensitive* to the initial choice of $a_1 = a$. Many single-variable calculus textbooks have an introduction to all of this for $n = 1$; e.g., see [THW, Sec. 4.7].

We illustrate the power of the multivariable Newton's method by revisiting the three examples above.

Example 18.5.6. In Example 18.5.1, we sought a zero of $\mathbf{f}(x, y) = (x^3 + xy + 2y^3 - 3, x^2y + xy^2 - y)$. The derivative matrix of this function, evaluated at any point, was computed in Example 13.5.5 and equals

$$(D\mathbf{f})(x, y) = \begin{bmatrix} 3x^2 + y & x + 6y^2 \\ 2xy + y^2 & x^2 + 2xy - 1 \end{bmatrix}. \tag{18.5.3}$$

Let us run Newton's method beginning at $\mathbf{a} = (0, 1)$ and see what happens.

Set $\mathbf{a}_1 = (0, 1)$, so $\mathbf{f}(\mathbf{a}_1) = (-1, -1)$. To go to the next step we use (18.5.3) to calculate

$$(D\mathbf{f})(\mathbf{a}_1) = \begin{bmatrix} 1 & 6 \\ 1 & -1 \end{bmatrix}, \text{ so } ((D\mathbf{f})(\mathbf{a}_1))^{-1} = \begin{bmatrix} 1/7 & 6/7 \\ 1/7 & -1/7 \end{bmatrix}.$$

Hence,

$$\mathbf{a}_2 = \mathbf{a}_1 - ((D\mathbf{f})(\mathbf{a}_1))^{-1}\mathbf{f}(\mathbf{a}_1) = \begin{bmatrix}0\\1\end{bmatrix} - \begin{bmatrix}1/7 & 6/7\\1/7 & -1/7\end{bmatrix}\begin{bmatrix}-1\\-1\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix} - \begin{bmatrix}-1\\0\end{bmatrix} = \begin{bmatrix}1\\1\end{bmatrix}$$

and then we compute $\mathbf{f}(\mathbf{a}_2) = (1,1)$. (That $\mathbf{f}(\mathbf{a}_2) = \mathbf{a}_2$ is a fluke, with no cosmic significance.) Now feeding $\mathbf{a}_2$ into (18.5.3) yields

$$(D\mathbf{f})(\mathbf{a}_2) = \begin{bmatrix}4 & 7\\3 & 2\end{bmatrix}, \text{ so } ((D\mathbf{f})(\mathbf{a}_2))^{-1} = \begin{bmatrix}-2/13 & 7/13\\3/13 & -4/13\end{bmatrix}.$$

Hence,

$$\mathbf{a}_3 = \mathbf{a}_2 - ((D\mathbf{f})(\mathbf{a}_2))^{-1}\mathbf{f}(\mathbf{a}_2) = \begin{bmatrix}1\\1\end{bmatrix} - \begin{bmatrix}-2/13 & 7/13\\3/13 & -4/13\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}1\\1\end{bmatrix} - \begin{bmatrix}5/13\\-1/13\end{bmatrix} = \begin{bmatrix}8/13\\14/13\end{bmatrix}$$

and then we calculate $\mathbf{f}(\mathbf{a}_3) = \mathbf{f}(8/13, 14/13) = (865/2197, 98/2197) \approx (0.3937187, 0.0446063)$.

At this point our long-suffering computer has had a lot of time to recover from its gambling experience in Section 16.2, so it is time to wake it up and call it back into action to continue running the steps of Newton's method. It springs to life and tells us that

$$\mathbf{a}_4 = \left(\frac{493367}{338052}, \frac{118892}{84513}\right)$$

with $\mathbf{f}(\mathbf{a}_4) = (7.729922331232\ldots, 4.4779445052955\ldots)$, and

$$\mathbf{a}_5 = \left(\frac{4163110851205464514415935765}{4237742372856894696580730802}, \frac{781166188908764911625281240}{706290395476149116096788467}\right)$$

with $\mathbf{f}(\mathbf{a}_5) = (1.74051781942412\ldots, 1.163107848685\ldots)$. We then tell the computer to stop showing off with the huge fractions, so it switches to decimal mode and kindly provides the following output:

k	$\mathbf{a}_k$
4	$(1.45944115106551654775\ldots, 1.40678948800776211943\ldots)$
5	$(0.98238884880556885346\ldots, 1.10601275893343829780\ldots)$
6	$(0.68009543839836081045\ldots, 1.04221078768332074644\ldots)$
7	$(0.61510603804391191781\ldots, 1.02420940858000089205\ldots)$
8	$(0.61155543524094804067\ldots, 1.02365061540338632705\ldots)$
9	$(0.61154729359324942964\ldots, 1.02364921627629839875\ldots)$
10	$(0.61154729354900381091\ldots, 1.02364921626903717013\ldots)$

and

k	$\mathbf{f}(\mathbf{a}_k)$
4	$(7.729922331232\ldots, 4.477944505295\ldots)$
5	$(1.740517819424\ldots, 1.163107848685\ldots)$
6	$(0.287472861154\ldots, 0.1785646922178\ldots)$
7	$(0.011527508214\ldots, 0.008555046538\ldots)$
8	$(0.000027121317\ldots, 0.000019600872\ldots)$
9	$(0.00000000145\ldots, 0.0000000001063\ldots)$
10	extremely small, around 10^{-21}

The points $\mathbf{a}_k$ stabilize well for $k \geq 8$ and $\mathbf{f}(\mathbf{a}_k)$ is close to $(0,0)$ beginning at $k = 8$, with $\mathbf{f}(\mathbf{a}_9)$ rather close to $(0,0)$. Since $\mathbf{f}(\mathbf{a}_{10}) \approx (0,0)$ to 21 decimal digits, $\mathbf{a}_{10}$ is an extraordinarily good approximate solution to $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. (For perspective on 21 decimal digits of accuracy, NASA uses π to an accuracy of 15 decimal digits for interplanetary space probes, such as the *Voyager* 1 spacecraft that is around

12.5 billion miles away. So the approximate vanishing of $\mathbf{f}(\mathbf{a}_{10})$ is more than sufficient for any current application.) ■

Example 18.5.7. Let's return to the area of inverse kinematics within robotics that we mentioned in Example 18.5.2. This will provide a further application of the motivating idea in (18.5.1) (resting on the linear approximation property of derivative matrices) for the definition of the multivariable Newton's method in (18.5.2). We focus on the situation with a planar robotic system in Example 13.5.11 to avoid complicated equations and input from physics; this special case will illustrate a mathematical insight with derivative matrices and their inverses that is relevant to rather general situations.

In notation from Example 13.5.11, the linear approximation property of derivative matrices yields

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \approx \Delta\mathbf{p} \approx (D\mathbf{p})(\vec{\theta})\,\Delta\vec{\theta} = \begin{bmatrix} \partial x/\partial\theta_1 & \partial x/\partial\theta_2 \\ \partial y/\partial\theta_1 & \partial y/\partial\theta_2 \end{bmatrix} \begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \end{bmatrix}.$$

The partial derivatives of x and y with respect to the θ_j's are computed using the expressions for x and y in terms of the θ_j's in (13.5.5), from which we obtain

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \approx \begin{bmatrix} -L_1\sin(\theta_1) - L_2\sin(\theta_1+\theta_2) & -L_2\sin(\theta_1+\theta_2) \\ L_1\cos(\theta_1) + L_2\cos(\theta_1+\theta_2) & L_2\cos(\theta_1+\theta_2) \end{bmatrix} \begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \end{bmatrix}. \tag{18.5.4}$$

We are given the left side (the desired small change in the position $\mathbf{p}$ of the tip), and we want to solve for the small change in angles to (approximately) achieve that. Exactly as in the reasoning with (18.5.1) which motivated Newton's method, we use the inverse of the 2×2 derivative matrix in (18.5.4). A bit of algebra and the addition law for sine gives that the determinant of that matrix is $L_1L_2\sin((\theta_1+\theta_2)-\theta_1) = L_1L_2\sin(\theta_2)$, so the formula for the inverse of a 2×2 matrix yields

$$\begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \end{bmatrix} = \frac{1}{L_1L_2\sin(\theta_2)} \begin{bmatrix} L_2\cos(\theta_1+\theta_2) & L_2\sin(\theta_1+\theta_2) \\ -L_1\cos(\theta_1) - L_2\cos(\theta_1+\theta_2) & -L_1\sin(\theta_1) - L_2\sin(\theta_1+\theta_2) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}.$$

But wait! What if the derivative matrix is not invertible? This corresponds to the vanishing of the determinant $L_1L_2\sin(\theta_2)$, so $\theta_2 = 0, \pi$. These are exactly the positions in which the robotic system is either fully stretched out ($\theta_2 = 0$) or is entirely folded up ($\theta_2 = \pi$), in which case a degree of freedom has been lost: the tip cannot be moved slightly in a direction along the line of the first arm.

That missing degree of freedom says (from such positions) we *cannot* achieve any desired small change in the position of the tip with a small change in the angles. This physical problem is detected mathematically (as above) through the non-invertibility of the derivative matrix for a configuration of the robotic arm. This is an instance of a phenomenon called "kinematic (or robotic) singularity" (recall that a square matrix is called "singular" when it is not invertible). In such configuations a robot gets stuck or behaves in unpredictable ways, and (in situations more intricate than the planar arm above) these are systematically found through a study of derivative matrices. The task of "singularity avoidance" in robotics involves algorithms to keep the robot away from these positions. ■

Example 18.5.8. For the application to GPS in Example 18.5.3, we need to solve a non-linear system of several equations in x, y, z, t, as well as some other non-linear equations. The usefulness of GPS (especially on a moving vehicle such as a car or airplane) requires that good approximate solutions are computed very rapidly. In practice, Newton's method is used to very rapidly (and accurately) solve such non-linear systems of equations. ■

Example 18.5.9. To find critical points as in Example 18.5.4 by applying Newton's method to a "first derivative" gradient function $\mathbf{f} = \nabla g$, the derivative matrix $(D\mathbf{f})(\mathbf{a})$ encodes the information of a "multivariable second derivative" of g called its *Hessian* (introduced in Section 25.2). In Appendix I we develop this application in detail and illustrate that its running time can be a big improvement on gradient descent. This underlies why the algorithm XGBoost wins many machine learning competitions. ■

Example 18.5.10. Another context where the idea behind the multivariable Newton's method arises is minimization for the magnitude of non-linear vector-valued functions; see Remark 22.5.7. ■

18.6. Equivalence of characterizations of invertibility. For those who are interested, we finish this chapter by coming back to an issue that was raised in Section 18.1 but not justified there, namely proving the equivalence of the following two conditions used to define invertibility for an $n \times n$ matrix A:

(a) The linear transformation $T_A : \mathbf{R}^n \to \mathbf{R}^n$ is invertible. Explicitly, for every $\mathbf{b} \in \mathbf{R}^n$ there is a unique $\mathbf{x} \in \mathbf{R}^n$ that solves the equation $A\mathbf{x} = \mathbf{b}$.
(b) There is an $n \times n$ matrix B with the property that $AB = \mathrm{I}_n$ and $BA = \mathrm{I}_n$ (in which case the linear transformation $T_B : \mathbf{R}^n \to \mathbf{R}^n$ is inverse to $T_A : \mathbf{R}^n \to \mathbf{R}^n$).

We first show that condition (a) implies condition (b). As in (a), suppose that the linear transformation $L = T_A$ is invertible. Its inverse function $M : \mathbf{R}^n \to \mathbf{R}^n$ is defined like this:

$$M(\mathbf{b}) \text{ is the unique } \mathbf{x} \in \mathbf{R}^n \text{ that satisfies } L\mathbf{x} = \mathbf{b}.$$

We claim M is actually a *linear transformation*. To check M is linear, pick $\mathbf{b}_1, \mathbf{b}_2 \in \mathbf{R}^n$ and scalars c_1, c_2, so we want to show

$$M(c_1\mathbf{b}_1 + c_2\mathbf{b}_2) \overset{?}{=} c_1 M(\mathbf{b}_1) + c_2 M(\mathbf{b}_2).$$

To verify this equality, we first use the invertibility of L to ensure that there is a unique vector $\mathbf{x}_1 \in \mathbf{R}^n$ satisfying $L(\mathbf{x}_1) = \mathbf{b}_1$ and a unique vector $\mathbf{x}_2 \in \mathbf{R}^n$ satisfying $L(\mathbf{x}_2) = \mathbf{b}_2$. By definition of M, we have $M(\mathbf{b}_1) = \mathbf{x}_1$ and $M(\mathbf{b}_2) = \mathbf{x}_2$, so the desired formula encoding linearity of M amounts to

$$M(c_1\mathbf{b}_1 + c_2\mathbf{b}_2) \overset{?}{=} c_1\mathbf{x}_1 + c_2\mathbf{x}_2.$$

By definition of M, the left side is the unique vector in $\mathbf{R}^n$ carried by L onto the vector $c_1\mathbf{b}_1 + c_2\mathbf{b}_2$. Thus, to show it is equal to the right side, we just have to check that the vector $c_1\mathbf{x}_1 + c_2\mathbf{x}_2 \in \mathbf{R}^n$ actually is carried by L onto $c_1\mathbf{b}_1 + c_2\mathbf{b}_2$.

To summarize, the asserted linearity of M comes down to verifying

$$L(c_1\mathbf{x}_1 + c_2\mathbf{x}_2) = c_1\mathbf{b}_1 + c_2\mathbf{b}_2.$$

But the right side is $c_1 L(\mathbf{x}_1) + c_2 L(\mathbf{x}_2)$ and we know

$$L(c_1\mathbf{x}_1 + c_2\mathbf{x}_2) = c_1 L(\mathbf{x}_1) + c_2 L(\mathbf{x}_2)$$

precisely because L is linear! This establishes the desired equality and so completes the verification of the claim that the inverse function $M : \mathbf{R}^n \to \mathbf{R}^n$ of L is linear. Hence, $M = T_B$ for some $n \times n$ matrix B. We will show that this B works as in condition (b).

By the very meaning of "inverse function", $M \circ L : \mathbf{R}^n \to \mathbf{R}^n$ carries each vector to itself and likewise for $L \circ M : \mathbf{R}^n \to \mathbf{R}^n$. But $M \circ L = T_B \circ T_A$ corresponds to the matrix BA (as was seen in our discussion relating matrix products to composition of linear transformations in Section 14.3), and likewise $L \circ M = T_A \circ T_B$ corresponds to the matrix AB. In other words, $M \circ L = T_{BA}$ and $L \circ M = T_{AB}$, so $T_{BA} : \mathbf{R}^n \to \mathbf{R}^n$ carries each vector to itself and $T_{AB} : \mathbf{R}^n \to \mathbf{R}^n$ carries each vector to itself. This says that $BA = \mathrm{I}_n$ because I_n is the matrix of the transformation that sends any input to itself as output, and similarly $AB = \mathrm{I}_n$. This completes the proof that (a) implies (b).

Now we show (b) implies (a). That is, assume there exists an $n \times n$ matrix B as in condition (b), and we shall show that for every $\mathbf{b} \in \mathbf{R}^n$ the vector equation $A\mathbf{x} = \mathbf{b}$ has a unique solution $\mathbf{x} \in \mathbf{R}^n$. Given such a solution vector $\mathbf{x}$, if we multiply both sides of the equation

$$A\mathbf{x} = \mathbf{b}$$

by B then we obtain $B(A\mathbf{x}) = B\mathbf{b}$. The left side is equal to $(BA)\mathbf{x}$, and $BA = \mathrm{I}_n$ due to how B was chosen, so $B(A\mathbf{x}) = (BA)\mathbf{x} = \mathrm{I}_n\mathbf{x} = \mathbf{x}$. We conclude that

$$\mathbf{x} = B\mathbf{b}.$$

In other words, the only *possible* solution to $A\mathbf{x} = \mathbf{b}$ is $\mathbf{x} = B\mathbf{b} \in \mathbf{R}^n$. But does $B\mathbf{b}$ really solve this vector equation? Indeed it does, because the product AB in the *other* order is equal to I_n:

$$A(B\mathbf{b}) = (AB)\mathbf{b} = \mathrm{I}_n\mathbf{b} = \mathbf{b}.$$

Thus, we have proved that condition (a) holds when condition (b) holds.

18.7. A visual interpretation of the 2-dimensional Newton's method. For a function $\mathbf{f} : \mathbf{R}^2 \to \mathbf{R}^2$ with component functions $f_1, f_2 : \mathbf{R}^2 \to \mathbf{R}$ (i.e., $\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{bmatrix}$) and a point $\begin{bmatrix} a \\ b \end{bmatrix} \in \mathbf{R}^2$, running a step of Newton's method yields as the output

$$\begin{bmatrix} a \\ b \end{bmatrix} - ((D\mathbf{f})(a,b))^{-1} \begin{bmatrix} f_1(a,b) \\ f_2(a,b) \end{bmatrix}. \tag{18.7.1}$$

We want to give a geometric interpretation of this construction that is similar in spirit to what was seen in the 1-dimensional case in Figure 18.5.1. In the latter case we used a tangent line to the graph of a single function $f(x)$ and saw where it crosses the x-axis. For the 2-dimensional case, we will consider tangent planes to surface graphs and there are *two* functions rather than one, namely f_1 and f_2; we will use the graph of each of these.

Here are analogous descriptions of Newton's method in the one-variable and two-variable cases:

- For the single-variable case (see Figure 18.5.1), if a is an approximate solution to $f(x) = 0$ and if $f'(a) \neq 0$ then at the point on the graph $y = f(x)$ in $\mathbf{R}^2$ with x-coordinate a, the tangent line there meets the x-axis at a point $(b, 0)$, with b the next approximate solution to $f(x) = 0$ in the method.
- For the two-variable case, if (a_1, a_2) is an approximate solution to $\mathbf{f}(x,y) = \mathbf{0}$ and if $(D\mathbf{f})(a_1, a_2)$ is invertible then at the points on the two graphs $z = f_1(x,y)$ and $z = f_2(x,y)$ in $\mathbf{R}^3$ with (x,y)-coordinate (a_1, a_2), the tangent planes there meet along a line that in turn meets the xy-plane at a point $(b_1, b_2, 0)$, with (b_1, b_2) the next approximate solution to $\mathbf{f}(x,y) = \mathbf{0}$ in the method.

Here is what is going algebraically in the visualization shown here to solve a pair of simultaneous (typically non-linear) equations $f_1(x,y) = 0$ and $f_2(x,y) = 0$. (The video uses the notation $r_j(u_1, u_2)$ rather than $f_j(x,y)$, and calls such values "residuals.") Consider the surface graphs $S_1 = \{z = f_1(x,y)\}$ (blue) and $S_2 = \{z = f_2(x,y)\}$ (red). The vertical line through $(a, b, 0)$ meets S_1 in the point $\mathbf{p}_1 = (a, b, f_1(a,b))$, and the same line meets S_2 in the point $\mathbf{p}_2 = (a, b, f_2(a,b))$. At each of these points, the corresponding surface has a tangent plane: at $\mathbf{p}_1$ on S_1 call the tangent plane P_1 and at $\mathbf{p}_2$ on S_2 call the tangent plane P_2. Under the necessary assumption for Newton's method that $(D\mathbf{f})(a,b)$ is invertible, we will show that P_1 and P_2 meet exactly along a line L, and that L meets the xy-plane in exactly one point, and that point is (18.7.1)!

To give a wider geometric context for what is going on in this visualization, we first note that two planes in $\mathbf{R}^3$ (not necessarily through the origin) typically meet along a line: see Figure 21.4.1 for a typical scenario. Likewise, a typical line in space (not necessarily through the origin) meets the xy-plane in exactly one point. So the typical visualization for two planes along with the xy-plane as a third plane is as shown in Figure 18.3.1. We claim that if $(D\mathbf{f})(a,b)$ is invertible then the three planes P_1, P_2, and the xy-plane indeed behave as in the typical situation in Figure 18.3.1, with the common point of overlap of all three planes being $(a', b', 0)$ where $\begin{bmatrix} a' \\ b' \end{bmatrix}$ is given by (18.7.1).

Now we shall show that the equations for P_1 and P_2 have a unique common solution in the xy-plane. The xy-plane is given by the equation $z = 0$, so our algebraic task is to show that among points of the form $(x, y, 0)$, exactly one such point satisfies the equations for both of the tangent planes P_1 and P_2. We also need to show that this point is $(a', b', 0)$ when $\begin{bmatrix} a' \\ b' \end{bmatrix}$ is given by (18.7.1). This will all be done in the following algebraic work.

It is time to bring out the equations for the tangent planes P_1 and P_2. By (11.2.3), the tangent plane P_1 to the surface graph $S_1 = \{z = f_1(x,y)\}$ at the point $(a, b, f_1(a,b))$ is

$$z = f_1(a,b) + (\partial f_1/\partial x)(a,b)(x-a) + (\partial f_1/\partial y)(a,b)(y-b)$$

and the tangent plane P_2 to the surface graph $S_2 = \{z = f_2(x,y)\}$ at the point $(a, b, f_2(a,b))$ is

$$z = f_2(a,b) + (\partial f_2/\partial x)(a,b)(x-a) + (\partial f_2/\partial y)(a,b)(y-b).$$

Setting z to be 0 in the equations for both P_1 and P_2, our goal is to show that the pair of equations

$$\begin{aligned} 0 &= f_1(a,b) + (\partial f_1/\partial x)(a,b)(x-a) + (\partial f_1/\partial y)(a,b)(y-b), \\ 0 &= f_2(a,b) + (\partial f_2/\partial x)(a,b)(x-a) + (\partial f_2/\partial y)(a,b)(y-b) \end{aligned}$$

has a unique simultaneous solution $\begin{bmatrix} a' \\ b' \end{bmatrix}$ that moreover is given by (18.7.1).

Let's reformulate these two scalar equations as a single matrix-vector equation:

$$\begin{bmatrix} (\partial f_1/\partial x)(a,b) & (\partial f_1/\partial y)(a,b) \\ (\partial f_2/\partial x)(a,b) & (\partial f_2/\partial y)(a,b) \end{bmatrix} \begin{bmatrix} x-a \\ y-b \end{bmatrix} = \begin{bmatrix} -f_1(a,b) \\ -f_2(a,b) \end{bmatrix} = -\begin{bmatrix} f_1(a,b) \\ f_2(a,b) \end{bmatrix} = -\mathbf{f}(a,b).$$

The matrix on the left is the derivative matrix $(D\mathbf{f})(a,b)$, and to make sense of Newton's method we have to assume that this derivative matrix is invertible. But then we can multiply both sides on the left by the inverse matrix to arrive at the *equivalent* formulation of the equations as saying

$$\begin{bmatrix} x-a \\ y-b \end{bmatrix} = ((D\mathbf{f})(a,b))^{-1}(-\mathbf{f}(a,b)) = -((D\mathbf{f})(a,b))^{-1}\mathbf{f}(a,b).$$

The left side is $\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} a \\ b \end{bmatrix}$, so adding $\begin{bmatrix} a \\ b \end{bmatrix}$ to both sides yields

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} - ((D\mathbf{f})(a,b))^{-1}\mathbf{f}(a,b)$$

as the unique solution. This shows the desired geometric property that the overlap of P_1 and P_2 meets the xy-plane in exactly one point, and the formula we have obtained for that point is exactly the output of one step of Newton's method on the input (a,b).

The following result, applied to the tangent planes P_1 and P_2, links the preceding algebraic work to the desired geometric interpretation of Newton's method.

Proposition 18.7.1. If two planes P and P' in $\mathbf{R}^3$ (not necessarily passing through the origin) have overlap that meets the xy-plane in exactly one point $\mathbf{p}$ then P and P' meet along exactly a line and this line meets the xy-plane in exactly $\mathbf{p}$.

Remark 18.7.2. The hypotheses in this result hold for the pair of tangent planes P_1 and P_2 using the point $\mathbf{p}$ given by (18.7.1).

PROOF. Since the planes P and P' in $\mathbf{R}^3$ meet in at least one point (namely, they have the point $\mathbf{p}$ in common), they either meet along exactly a line or are the same plane. First, we rule out the possibility $P = P'$. If the planes coincide then their overlap is that same plane, but two planes in $\mathbf{R}^3$ (such as $P = P'$ and the xy-plane) cannot meet in just a single point (two planes in $\mathbf{R}^3$ which meet at all must at least meet along a line). But we are assuming that the overlap of P and P' meets the xy-plane at precisely one point (so not a line!). Hence, the possibility $P = P'$ is ruled out.

But P and P' meet in $\mathbf{R}^3$ in at least the point $\mathbf{p}$, so these distinct planes cannot be parallel and hence they indeed meet along exactly a line as desired. That line in turn meets the xy-plane in exactly one point by our assumptions on where the overlap of P and P' meets the xy-plane. ☐

Chapter 18 highlights (links to highlights in and next chapters)

Notation	Meaning	Location in text
A^{-1}	inverse of an $n \times n$ matrix A	Prop. 18.1.5, Def. 18.1.6
$\det(A)$ (for 2×2 matrix A)	determinant of A	Rem. 18.2.6 and preceding few paragraphs

Concept	Meaning	Location in text
invertible function f	$f(x) = b$ has unique solution for each b; i.e., f has an unambiguous inverse	Examples 18.1.1–18.1.4
inverse function	for a function f, this is a function g in the opposite direction for which $g(f(x)) = x$ for all x and $f(g(y)) = y$ for all y	Example 18.1.1
invertible linear transformation T	$T(\mathbf{x}) = \mathbf{b}$ has unique solution for each $\mathbf{b}$	Proposition 18.1.5(a)
invertible matrix, inverse matrix	$n \times n$ matrix A for which $AB = \mathrm{I}_n = BA$ for some B (called "inverse" of A)	Definition 18.1.6
determinant ($n = 2$)	for $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, it is $ad - bc$	Definition 18.2.4

Result	Meaning	Location in text
invertibility of $n \times n$ matrix A in product sense same as invertibility of T_A	there is B with $AB = \mathrm{I}_n = BA$ precisely when $T_A : \mathbf{R}^n \to \mathbf{R}^n$ is invertible	Proposition 18.1.5
det criterion for invertibility ($n = 2$)	A^{-1} exists precisely when $\det(A) \neq 0$	Example 18.2.1, Remark 18.2.3
area meaning of $\lvert\det(A)\rvert$ ($n = 2$)	$T_A : \mathbf{R}^2 \to \mathbf{R}^2$ distorts area by $\lvert\det(A)\rvert$	Remark 18.2.6
invertibility of $n \times n$ matrix A can be checked "on one side"	$AB = \mathrm{I}_n$ precisely when $BA = \mathrm{I}_n$	Theorem 18.1.8
$A\mathbf{x} = \mathbf{b}$ influenced by invertibility of A	if A invertible then $A^{-1}\mathbf{b}$ is unique solution; if A not invertible either no solution or infinitely many (depending on $\mathbf{b}$)	Theorem 18.3.3
inversion swaps order of multiplication	if A, B invertible then so is AB, with $(AB)^{-1} = B^{-1}A^{-1}$	(ii) in big box at start of Sec. 18.4
Newton's method for zeros	technique to approximate a solution to $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ for non-linear $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^n$ ("n equations in n unknowns")	(18.5.2)

Skill	Location in text
for general 2×2 or diagonal $n \times n$ matrix A, check invertibility and find A^{-1}	(18.1.1), (18.2.2), Ex. 18.2.2
for $n \times n$ matrix A, given A^{-1} use it to solve $A\mathbf{x} = \mathbf{b}$	Example 18.3.1
in $n \times n$ case, relate $(AB)^{-1}$ to A^{-1} and B^{-1}	Example 18.4.3
be aware of subtleties with matrix inverse	big box at start of Sec. 18.4
run 1 step of Newton's method for zeros when $n = 2$	Example 18.5.6

Part IV

Further matrix algebra and linear systems

"Rarely does anything understood deeply turn out to be useless, even though it may not serve the purpose you had hoped."

K. Smith, Keeler Professor of Mathematics (University of Michigan)

"Of course it is happening inside your head [. . .], but why on earth should that mean it is not real?"

A. Dumbledore [Ro]

Overview of Part IV

This Part of the book, comprising Chapters 19–22, introduces additional notions in matrix algebra, and discusses applications to linear systems; i.e., collections of equations that look like this:

$$3x - 4y + z - 7w = 1, \quad 2x - y - z + w = 0, \quad x + y - z - 2w = 5.$$

The new concepts we introduce in matrix algebra are of two types: algebraic and geometric. In Chapter 19 we introduce the algebraic notion of *linear independence* (a useful way of expressing that a collection of vectors has "no redundancy") and the related geometric algorithm called the *Gram–Schmidt process*. This algorithm is a generalization of the higher-dimensional Pythagorean Theorem (Theorem 2.3.1), and it solves the puzzle (which arises whenever trying to use the method in Theorem 6.2.1 to compute projections to a subspace of dimension ≥ 2) of how to build an orthogonal basis of a general linear subspace of $\mathbf{R}^n$. It is also pervasive in applications of linear algebra (e.g., the QR-decomposition in Chapter 22), as well as in the arguments in the first section of the optional Appendix B that justify many intuitive expectations about "dimension" (which we stated in earlier parts of this book and have used throughout). For example, the Gram–Schmidt process provides us with a systematic way of determining when the span of a given collection of k nonzero n-vectors has dimension equal to k (rather than $< k$).

In Chapter 20 we introduce and explore the algebraic notion of *matrix transpose* and use it to define the distinguished algebraic class of *symmetric* $n \times n$ matrices as well as describe the distinguished geometric class of *orthogonal* linear transformations $\mathbf{R}^n \to \mathbf{R}^n$, each of which plays a role later in the book. For instance, properties of symmetric matrices underlie the multivariable second-derivative test in Chapter 26, and the geometry of orthogonal transformations underlies the *singular value decomposition* (SVD) (a special case of which is called *principal component analysis* (PCA)) in Section 27.3. The implementation of SVD has important applications in many fields: robotics, gene expression analysis, machine learning, image compression, web search, quantum information, ... In Chapter 20 we also introduce the algebraic concept of *quadratic form*, which arises naturally in the study of energy for physical systems with many parts and whose connection to symmetric matrices will be important for understanding both the multivariable second-derivative test as well as SVD.

In Chapters 21–22 we explore systems of linear equations using the algebraic and geometric tools developed so far. (Such "linear systems" in many variables – far beyond just 3 variables – arise *everywhere*, as we illustrate in Section 21.1.) Although finding an explicit numerical solution to a linear system is a job usually best left to a computer, at least if the equations involve more than a few variables, there is a conceptual framework behind the properties and behavior of linear systems that is *important to know, even when using software*. Likewise, when aiming to explore the *qualitative behavior* of solutions to large systems of linear equations, having "geometric intuition" for large linear systems is very valuable. Understanding concepts underlying the calculations is what enables one to do novel things.

19. Linear independence and the Gram–Schmidt process

The notions of dimension and basis for a linear subspace V of $\mathbf{R}^n$ were introduced in Chapters 4 and 5, but we have only seen how to verify that a given spanning set of k nonzero vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$ for V is of minimal size (and so is a basis) when $k \leq 3$ (so the Dimension Criterion in Section 5.1 applies) or when $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ is a collection of orthogonal vectors (so Theorem 5.2.2 applies).

In this chapter, we will show how to construct a basis for any linear subspace V of $\mathbf{R}^n$ starting from a spanning set for V. More precisely, we will construct an orthogonal basis for V; the only input will be a spanning set for V. As a bonus, we will be able to establish several reasonable-sounding geometric properties of bases of subspaces that are suggested by a gridline picture in $\mathbf{R}^3$ (and that we would like to hold in general).

The procedure for building an orthogonal basis from a spanning set is called the *Gram–Schmidt process*, and it has been seen in the case of 2 nonzero vectors $\mathbf{x}, \mathbf{y}$ in Theorem 7.1.1. (Strictly speaking, the procedure in Theorem 7.1.1 requires first checking that $\mathbf{x}$ and $\mathbf{y}$ are not scalar multiples of each other, so their span is really a plane. When that fails, the span is just a line.)

One merit of knowing an orthogonal basis of a plane is that it can be used to compute projections into a plane, as we have seen and used in Chapter 7. In a wide array of contexts (such as in signal processing, economics, and principal component analysis – one of the key algorithms in data science) it is useful to be able to compute projections into a *general* subspace V of $\mathbf{R}^n$, and we know how to compute $\mathbf{Proj}_V$ in terms of an orthogonal basis of V.

By the end of this chapter, you should be able to do the following for a collection of nonzero vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$ in $\mathbf{R}^n$ with span V:

- use the Gram–Schmidt process to compute an orthogonal basis of V and compute $\dim V$;
- determine whether or not $\dim(\operatorname{span}(\mathbf{v}_1, \dots, \mathbf{v}_k)) = k$;
- relate the $\mathbf{v}_i$'s being a basis of their span to the $\mathbf{v}_i$'s being linearly independent.

19.1. Linear independence. In Definition 4.2.4, we defined the dimension of a nonzero linear subspace V of $\mathbf{R}^n$ to be the minimal size of a spanning set. However, if we are given a spanning set $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ for V, it isn't apparent how to determine whether that spanning set is of minimal size among all possible spanning sets for V. Revisiting Example 5.2.4 for some guidance, consider the three vectors

$$\mathbf{v}_1 = \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 0 \\ 2 \\ -1 \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} 2 \\ -4 \\ 3 \end{bmatrix}$$

in $\mathbf{R}^3$. We saw in Example 5.2.4 that these lie inside a common plane in $\mathbf{R}^3$ (so their span is at most 2-dimensional rather than 3-dimensional), and that one of these $\mathbf{v}_i$'s belongs to the span of the others:

$$\mathbf{v}_3 = (2/3)\mathbf{v}_1 - (5/3)\mathbf{v}_2$$

(as can be verified directly, and discovered by the procedure in Example 5.1.7).

More generally, for the span V of k vectors $\mathbf{v}_1, \dots, \mathbf{v}_k \in \mathbf{R}^n$ with $k > 1$, one reason $\dim V$ could fail to equal k (i.e., the given spanning set could fail to be of minimal size) is that there is "redundancy": perhaps

$$\mathbf{v}_k = c_1\mathbf{v}_1 + \cdots + c_{k-1}\mathbf{v}_{k-1}$$

for some scalars $c_1, \dots, c_{k-1}$ (i.e., $\mathbf{v}_k$ belongs to the span of the other $\mathbf{v}_j$'s). Generalizing (5.2.3), V is then spanned by $\mathbf{v}_1, \dots, \mathbf{v}_{k-1}$ because *any* linear combination of $\mathbf{v}_1, \dots, \mathbf{v}_k$ can be written in terms of

$\mathbf{v}_1, \dots, \mathbf{v}_{k-1}$: by substituting the expression for $\mathbf{v}_k$ in terms of the rest we get

$$\begin{aligned} a_1\mathbf{v}_1 + \cdots + a_k\mathbf{v}_k &= (a_1\mathbf{v}_1 + \cdots + a_{k-1}\mathbf{v}_{k-1}) + a_k(c_1\mathbf{v}_1 + \cdots + c_{k-1}\mathbf{v}_{k-1}) \\ &= \sum_{j=1}^{k-1} a_j\mathbf{v}_j + \sum_{j=1}^{k-1} a_kc_j\mathbf{v}_j \\ &= \sum_{j=1}^{k-1}(a_j + a_kc_j)\mathbf{v}_j. \end{aligned}$$

There is nothing special about $\mathbf{v}_k$: for any $1 \leq i \leq k$, if $\mathbf{v}_i$ is a linear combination of the rest then it can be dropped without affecting the span (so again $\dim V \leq k - 1 < k$).

This brings us to some questions, inspired by visualization in $\mathbf{R}^3$:

(i) if $\dim V < k$ then is that always "explained" by redundancy: *some* $\mathbf{v}_i$ belonging to the span of the others (a situation for which, as we saw above, it is always the case that $\dim V < k$)?

(ii) Assuming (i) is correct and that $\dim V < k$, remove from $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ some $\mathbf{v}_i$ that is in the span of the rest. If the resulting collection of $k - 1$ vectors has a member belonging to the span of the rest, remove that one and continue in this way. When this process stops, is what remains a basis for V? (If so, we will have found a basis for V *inside* any given spanning set for V.)

The answers to both questions turn out to be "yes", but to make the answers algorithmic we need a systematic procedure to built an orthogonal basis from a given spanning set. To begin, we need:

Definition 19.1.1. For $k > 1$, a collection of vectors $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ in $\mathbf{R}^n$ is called *linearly dependent* if some $\mathbf{v}_i$ belongs to the span of the others. Otherwise it is called *linearly independent* (i.e., no $\mathbf{v}_i$ belongs to the span of the others). A collection $\{\mathbf{v}\}$ consisting of a single vector is called *linearly dependent* when $\mathbf{v} = \mathbf{0}$ and is called *linearly independent* when $\mathbf{v} \neq \mathbf{0}$.

The visualization of linear independence is that each $\mathbf{v}_i$ contributes an "independent direction of motion" within the span of $\mathbf{v}_1, \dots, \mathbf{v}_k$, whereas linear dependence expresses "redundancy" for the purpose of spanning; see Figure 19.1.1. The separate treatment for $k = 1$ in Definition 19.1.1 may be annoying; a formulation treating all $k \geq 1$ uniformly will be given in Theorem 19.1.5.

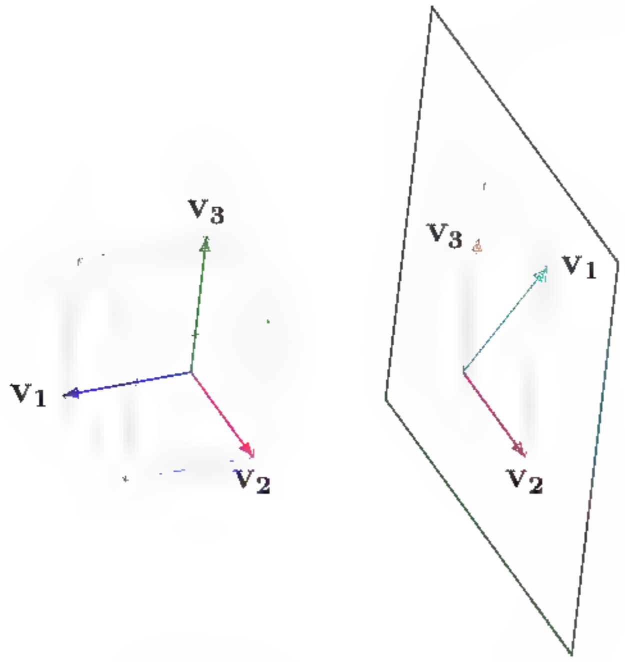


FIGURE 19.1.1. Two collections of three vectors $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ in $\mathbf{R}^3$; the first (edges of a cube with a corner at $\mathbf{0}$) is linearly independent and the second is linearly dependent.

Example 19.1.2. The vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ in Example 5.2.4 are linearly dependent due to the equality

$$\mathbf{v}_3 = (2/3)\mathbf{v}_1 - (5/3)\mathbf{v}_2$$

that we have seen above. Multiplying through by 3 and bringing the $\mathbf{v}_3$-term to the other side yields

$$2\mathbf{v}_1 - 5\mathbf{v}_2 - 3\mathbf{v}_3 = \mathbf{0}.$$

This latter equation can be "solved" for either of $\mathbf{v}_1$ or $\mathbf{v}_2$ in terms of the rest, by bringing the other vectors to the other side of the equation and dividing by the scalar multiplier:

$$2\mathbf{v}_1 = 5\mathbf{v}_2 + 3\mathbf{v}_3 \text{ implies } \mathbf{v}_1 = \frac{5}{2}\mathbf{v}_2 + \frac{3}{2}\mathbf{v}_3, \quad \text{and } -5\mathbf{v}_2 = -2\mathbf{v}_1 + 3\mathbf{v}_3 \text{ implies } \mathbf{v}_2 = \frac{2}{5}\mathbf{v}_1 - \frac{3}{5}\mathbf{v}_3.$$

One lesson we learn is that there is nothing special about $\mathbf{v}_3$ belonging to the span of $\mathbf{v}_1$ and $\mathbf{v}_2$ in this example: we can also write $\mathbf{v}_1$ in terms of $\mathbf{v}_2$ and $\mathbf{v}_3$, and also write $\mathbf{v}_2$ in terms of $\mathbf{v}_1$ and $\mathbf{v}_3$. ■

Example 19.1.3. In general when $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ is linearly dependent with $k > 1$, all we know (by definition) is that *some* $\mathbf{v}_i$ is a linear combination of the others. When such an expression $\mathbf{v}_i = \sum_{j \neq i} a_j \mathbf{v}_j$ has *all* coefficients a_j nonzero then we can write each $\mathbf{v}_j$ in terms of the other $\mathbf{v}$'s much as we did in Example 19.1.2. However, if some a_j vanishes then we cannot conclude anything about writing the corresponding $\mathbf{v}_j$ as a linear combination of the other $\mathbf{v}$'s and this may be impossible (so the lesson at the end of Example 19.1.2 shouldn't be misunderstood). For instance, consider the vectors

$$\mathbf{v}_1 = \begin{bmatrix} 3 \\ -1 \\ 2 \\ 4 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 0 \\ 2 \\ -1 \\ 1 \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} 2 \\ -4 \\ 3 \\ 1 \end{bmatrix}, \quad \mathbf{v}_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

One can check that $\mathbf{v}_1 = (5/2)\mathbf{v}_2 + (3/2)\mathbf{v}_3$ (this is closely related to Examples 5.2.4 and 19.1.2), so the entire collection $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$ is linearly dependent by using a coefficient of 0 for $\mathbf{v}_4$:

$$\mathbf{v}_1 = (5/2)\mathbf{v}_2 + (3/2)\mathbf{v}_3 + 0\mathbf{v}_4. \tag{19.1.1}$$

(Via the device of coefficients of 0, any finite collection of vectors containing a linearly dependent subcollection is itself linearly dependent.) We claim it is *impossible* to express $\mathbf{v}_4$ as a linear combination of $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ (i.e., $\mathbf{v}_4$ is *not* in the span of $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$). The fact that (19.1.1) can't be used to obtain such an expression (due to $\mathbf{v}_4$ having a vanishing coefficient there) doesn't imply this cannot be done; we need to analyze the situation more closely to determine if this is truly impossible or not.

First, we simplify our task: if there were *some* expression $a\mathbf{v}_1 + b\mathbf{v}_2 + c\mathbf{v}_3 = \mathbf{v}_4$ then via (19.1.1) we could plug $(5/2)\mathbf{v}_2 + (3/2)\mathbf{v}_3$ in place of $\mathbf{v}_1$ in this expression for $\mathbf{v}_4$ to obtain

$$\mathbf{v}_4 = a\mathbf{v}_1 + b\mathbf{v}_2 + c\mathbf{v}_3 = a((5/2)\mathbf{v}_2 + (3/2)\mathbf{v}_3) + b\mathbf{v}_2 + c\mathbf{v}_3 = ((5/2)a + b)\mathbf{v}_2 + ((3/2)a + c)\mathbf{v}_3,$$

so $\mathbf{v}_4$ would be in the span of $\mathbf{v}_2$ and $\mathbf{v}_3$. But if for some scalars x and y we really had an equality

$$\mathbf{v}_4 = x\mathbf{v}_2 + y\mathbf{v}_3$$

of 4-vectors then equating corresponding vector entries on both sides yields a system of 4 equations in x and y that we'll show has no solution (so there is no such expression for $\mathbf{v}_4$). The equations are

$$0 = 0x + 2y, \quad 0 = 2x - 4y, \quad 0 = -x + 3y, \quad 1 = x + y.$$

The first equation forces $y = 0$, and putting that into the second or third equation forces $x = 0$, but $(x, y) = (0, 0)$ violates the fourth equation. Hence, indeed $\mathbf{v}_4$ cannot be written in the form $a\mathbf{v}_1 + b\mathbf{v}_2 + c\mathbf{v}_3$. ■

Remark 19.1.4. If $\mathbf{v}_1, \dots, \mathbf{v}_k$ is a collection of linearly independent vectors, then all $\mathbf{v}_i$'s must be nonzero! Indeed, for $k = 1$ this is part of the definition of linear independence. If $k > 1$ and some $\mathbf{v}_i$ is equal to $\mathbf{0}$ then $\mathbf{v}_i = \sum_{j \neq i} 0\mathbf{v}_j$, expressing $\mathbf{v}_i$ as a (not so interesting) linear combination of the others.

It may seem mysterious how to determine if a given collection of nonzero vectors in $\mathbf{R}^n$ is linearly dependent or linearly independent, since usually we can't just guess how to express some $\mathbf{v}_i$ as a linear combination of the others (or see by inspection that this is impossible). We now give a criterion for linear independence, inspired by the calculations in Example 19.1.2, that we can apply in some special examples (and whose utility beyond special circumstances will be amply illustrated in Chapters 20 and 21).

Theorem 19.1.5. A collection of vectors $\mathbf{v}_1, \dots, \mathbf{v}_k \in \mathbf{R}^n$ is linearly independent precisely when the *only* collection of scalars $a_1, \dots, a_k$ for which

$$a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \cdots + a_k\mathbf{v}_k = \mathbf{0}$$

is $a_1 = 0, a_2 = 0, \dots, a_k = 0$. **In particular, any collection of mutually orthogonal nonzero vectors $\mathbf{w}_1, \dots, \mathbf{w}_k$ in $\mathbf{R}^n$ is linearly independent.**

Put another way, the collection of $\mathbf{v}_i$'s is linearly dependent precisely when there are coefficients $a_1, \dots, a_k$ not all equal to 0 for which $\sum_{i=1}^k a_i\mathbf{v}_i = \mathbf{0}$. (In this criterion for linear dependence we are demanding that *some* coefficient a_i isn't equal to 0, not that *all* the a_i's are nonzero.)

PROOF. The proof is an adaptation the calculations in Example 19.1.2 to show linear *dependence* is the same as the condition that the linear combination $\sum_{i=1}^k a_i\mathbf{v}_i$ vanishes for some collection of coefficients $a_1, \dots, a_k$ that aren't all equal to 0. At the end we address the orthogonality assertion.

If linear dependence holds and $k > 1$ then some $\mathbf{v}_i$ is in the span of the rest, which is to say $\mathbf{v}_i = \sum_{j \neq i} c_j\mathbf{v}_j$ for some scalars c_j (with $j \neq i$). But then bringing $\mathbf{v}_i$ to the other side, we have

$$\mathbf{0} = -\mathbf{v}_i + \sum_{j \neq i} c_j\mathbf{v}_j = (-1)\mathbf{v}_i + \sum_{j \neq i} c_j\mathbf{v}_j.$$

This is a vanishing linear combination for which the coefficient of $\mathbf{v}_i$ is $-1 \neq 0$. If linear dependence holds and $k = 1$ then $\mathbf{v}_1 = \mathbf{0}$, so $1\mathbf{v}_1 = \mathbf{0}$; this is a vanishing linear combination for which the coefficient of $\mathbf{v}_1$ is $1 \neq 0$.

On the other hand, suppose $\sum_{j=1}^k a_j\mathbf{v}_j = \mathbf{0}$ with some coefficient a_i not equal to 0. Suppose $k > 1$. Bringing the ith term to the other side yields

$$\sum_{j \neq i} a_j\mathbf{v}_j = -a_i\mathbf{v}_i.$$

Now we can multiply both sides by the scalar $-1/a_i$ (which makes sense since $a_i \neq 0$) to obtain

$$\sum_{j \neq i} (-a_j/a_i)\mathbf{v}_j = \mathbf{v}_i,$$

expressing $\mathbf{v}_i$ as a linear combination of the other $\mathbf{v}_j$'s. This establishes linear dependence when $k > 1$. Suppose $k = 1$, so $a_1\mathbf{v}_1 = \mathbf{0}$ with $a_1 \neq 0$ (since some coefficient is assumed to be nonzero and now there is only one coefficient). Then we can multiply both sides by $1/a_1$ (which makes sense since $a_1 \neq 0$) to obtain $\mathbf{v}_1 = \mathbf{0}$, so linear dependence holds (by its definition when $k = 1$).

Finally, suppose $\mathbf{w}_1, \dots, \mathbf{w}_k$ is a collection of mutually orthogonal nonzero n-vectors. To show linear independence, we assume $\sum_{i=1}^k a_i\mathbf{w}_i = \mathbf{0}$ for some scalars $a_1, \dots, a_k$ and aim to prove each a_j vanishes. Applying dot product against $\mathbf{w}_j$ to both sides yields $\sum_{i=1}^k a_i(\mathbf{w}_i \cdot \mathbf{w}_j) = \mathbf{0} \cdot \mathbf{w}_j = 0$. By orthogonality $\mathbf{w}_i \cdot \mathbf{w}_j = 0$ for all $i \neq j$, so the sum collapses to the term $i = j$: $a_j(\mathbf{w}_j \cdot \mathbf{w}_j) = 0$. But $\mathbf{w}_j \cdot \mathbf{w}_j = \|\mathbf{w}_j\|^2 > 0$ since $\mathbf{w}_j \neq \mathbf{0}$, so we can divide by that to get $a_j = 0$ as desired. □

Here are some illustrations of Theorem 19.1.5 in action.

Example 19.1.6. Consider the standard basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ of $\mathbf{R}^3$. Using the criterion in Theorem 19.1.5, these are linearly independent since the condition

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + a_3\mathbf{e}_3 = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

forces all a_i's to vanish. Alternatively, the $\mathbf{e}_j$'s are mutually orthogonal and nonzero, so they're linearly independent. ■

Example 19.1.7. In Example 4.1.13 we exhibited three vectors in $\mathbf{R}^5$ spanning a specific subspace W:

$$\mathbf{v}_1 = \begin{bmatrix} -1 \\ 1 \\ -2/3 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 1 \\ 0 \\ -1/3 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} -2 \\ 0 \\ 1/3 \\ 0 \\ 1 \end{bmatrix}.$$

We claim that this collection is linearly independent. Using the criterion in Theorem 19.1.5, we want to check that the only way we can have

$$a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + a_3\mathbf{v}_3 = \mathbf{0}$$

is when the scalars a_1, a_2, a_3 all vanish.

Writing out everything explicitly,

$$a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + a_3\mathbf{v}_3 = \begin{bmatrix} -a_1 + a_2 - 2a_3 \\ a_1 \\ -2a_1/3 - a_2/3 + a_3/3 \\ a_2 \\ a_3 \end{bmatrix},$$

so by staring at the second, fourth, and fifth entries we see that the vanishing of this vector forces $a_1 = 0, a_2 = 0, a_3 = 0$ as desired. ■

The two preceding examples as applications of Theorem 19.1.5 were tractable because the vectors were very special: there were lots of entries equal to 0, enough so that when forming any possible linear combination, each scalar coefficient wound up being isolated in its own entry in the linear combination vector. We will rarely be so lucky in practice. Here is a more typical scenario to illustrate the shortcomings of Theorem 19.1.5 at our present state of knowledge:

Example 19.1.8. Consider the vectors

$$\mathbf{v}_1 = \begin{bmatrix} -2 \\ 1 \\ 5 \\ 3 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 4 \\ 3 \\ -3 \\ 1 \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} 13 \\ 6 \\ -15 \\ -2 \end{bmatrix}$$

in $\mathbf{R}^4$. Are these linearly dependent? By staring at the vectors, we don't see in any easy way that one of these $\mathbf{v}_i$'s is in the span of the others, but that doesn't mean no such expression is possible.

The criterion in Theorem 19.1.5 says that we should consider triples a_1, a_2, a_3 for which

$$a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + a_3\mathbf{v}_3 = \mathbf{0},$$

and determine whether this can happen with some $a_i \neq 0$. In terms of explicit vectors, this equation says

$$\begin{bmatrix} -2a_1 \\ a_1 \\ 5a_1 \\ 3a_1 \end{bmatrix} + \begin{bmatrix} 4a_2 \\ 3a_2 \\ -3a_2 \\ a_2 \end{bmatrix} + \begin{bmatrix} 13a_3 \\ 6a_3 \\ -15a_3 \\ -2a_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

or in other words

$$\begin{bmatrix} -2a_1 + 4a_2 + 13a_3 \\ a_1 + 3a_2 + 6a_3 \\ 5a_1 - 3a_2 - 15a_3 \\ 3a_1 + a_2 - 2a_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Equating corresponding entries on both sides, we get the following system of 4 equations in 3 unknowns.

$$\begin{aligned} -2a_1 + 4a_2 + 13a_3 &= 0 \\ a_1 + 3a_2 + 6a_3 &= 0 \\ 5a_1 - 3a_2 - 15a_3 &= 0 \\ 3a_1 + a_2 - 2a_3 &= 0 \end{aligned}$$

Does it have a solution (a_1, a_2, a_3) different from the uninteresting solution $(0, 0, 0)$? If so then linear dependence holds, and if not then linear independence holds.

It isn't at all clear by staring at this system of equations if it has a solution different from $(0, 0, 0)$. It turns out that there *are* other solutions, such as $(-3, 5, -2)$, as you can verify by direct substitution into the system of equations (though it shouldn't be at all apparent how we found that triple). In other words:

$$-3\mathbf{v}_1 + 5\mathbf{v}_2 - 2\mathbf{v}_3 = \mathbf{0}$$

(as can be verified by direct calculation). This allows us to solve for each $\mathbf{v}_i$ in terms of the others much as in Example 19.1.2, exhibiting linear dependence in several ways that you probably couldn't have guessed from staring at the definitions of $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$:

$$\mathbf{v}_1 = (5/3)\mathbf{v}_2 - (2/3)\mathbf{v}_3, \quad \mathbf{v}_2 = (3/5)\mathbf{v}_1 + (2/5)\mathbf{v}_3, \quad \mathbf{v}_3 = -(3/2)\mathbf{v}_1 + (5/2)\mathbf{v}_2.$$

(Please remember however the lesson from Example 19.1.3 that in general linear dependence for a collection of vectors $\mathbf{x}_1, \ldots, \mathbf{x}_k$ with $k > 1$ does not mean that *every* $\mathbf{x}_i$ is a linear combination of the others, but only that *some* $\mathbf{x}_i$ is a linear combination of the others.)

The upshot is that using Theorem 19.1.5 typically requires confronting a system of n equations in k unknowns, a task that looks quite formidable in general (and which we will learn how to approach in a systematic and efficient way in Chapter 22). ■

To circumvent the algebraic difficulties as in Example 19.1.8, we will next develop a *geometric* approach to computing the dimension of a linear subspace that will also determine without any reliance on luck or cleverness whether a given collection of vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k \in \mathbf{R}^n$ is linearly independent or linearly dependent. This involves solving the motivating problem posed at the start of this chapter (finding orthogonal bases of linear subspaces), so we now turn to that topic.

19.2. Orthogonal bases and Gram–Schmidt. The Gram–Schmidt process is an algorithm (that a computer can implement easily, and a person can carry out by hand when not too many vectors are involved) to find an orthogonal basis $\mathbf{w}_1, \mathbf{w}_2, \ldots$ of a nonzero subspace V of $\mathbf{R}^n$ when given a spanning set $\mathbf{v}_1, \ldots, \mathbf{v}_k$ of nonzero vectors in V.

When one has a basis for a subspace and it is an *orthogonal* basis, a variety of calculations are much easier to do (either by hand or by computer). For instance, we've seen already that having an orthogonal basis is quite useful for computing projections, which has many applications within mathematics (e.g., the geometry involved in computing the closest point to a line or a plane) and outside of mathematics (e.g., linear regression, signal processing, economics, principal component analysis, quantum mechanics, and much else). Thus, having an easily implemented algorithm for finding an orthogonal basis from a spanning set is very useful.

Remark 19.2.1. Let's briefly elaborate on the relevance in signal processing of finding an orthogonal basis. When electronic signals are communicated across a channel, there is a collection of basic signals out of which all others are built via linear combinations. The basic signals can be very "correlated" with each other, making computations in terms of the basis of basic signals rather messy. The merit of finding an orthogonal (or better yet, orthonormal) basis of the "signal space" is that it provides a collection of signals that are "uncorrelated" with each other (this expresses the orthogonality condition) and consequently much better suited to a wide variety of computations. In particular, analogues of the coefficient formula (19.2.1) below underlie the simplifications achieved in such computations.

We can't get into specific details here since in the mathematical model one has to replace the Euclidean spaces $\mathbf{R}^n$ we have been using with some "function spaces", and the analogue of the dot product is a certain integral. The resulting "squared distance" expresses the energy of a signal.

Remark 19.2.2. Having an orthogonal basis for a subspace is useful within the subject of linear algebra itself. For example, when V is a subspace of $\mathbf{R}^n$ given by the span of k vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ then any vector $\mathbf{v} \in V$ can be written as a linear combination

$$\mathbf{v} = a_1\mathbf{v}_1 + \cdots + a_k\mathbf{v}_k$$

but the coefficients $a_1, \ldots, a_k$ may be rather difficult to compute. However, if we can *find* an orthogonal basis $\{\mathbf{w}_1, \ldots, \mathbf{w}_m\}$ of V (possibly $m < k$), we can also write $\mathbf{v}$ as

$$\mathbf{v} = c_1\mathbf{w}_1 + \cdots + c_m\mathbf{w}_m$$

with coefficients $c_1, \ldots, c_m$ that are *easy* to compute via the Fourier formula in Theorem 5.3.6:

$$c_i = \frac{\mathbf{v} \cdot \mathbf{w}_i}{\mathbf{w}_i \cdot \mathbf{w}_i}. \tag{19.2.1}$$

Having discussed the merits of knowing an orthogonal basis of a subspace V of $\mathbf{R}^n$, we now describe the general procedure to find one when given just a spanning set of V. (The proof that the procedure below works as claimed is given in Section B.1.) Figure 19.2.1 illustrates the geometry of what the procedure does to 3 linearly independent vectors $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ in $\mathbf{R}^3$ to produce an orthogonal basis $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ of $\mathbf{R}^3$.

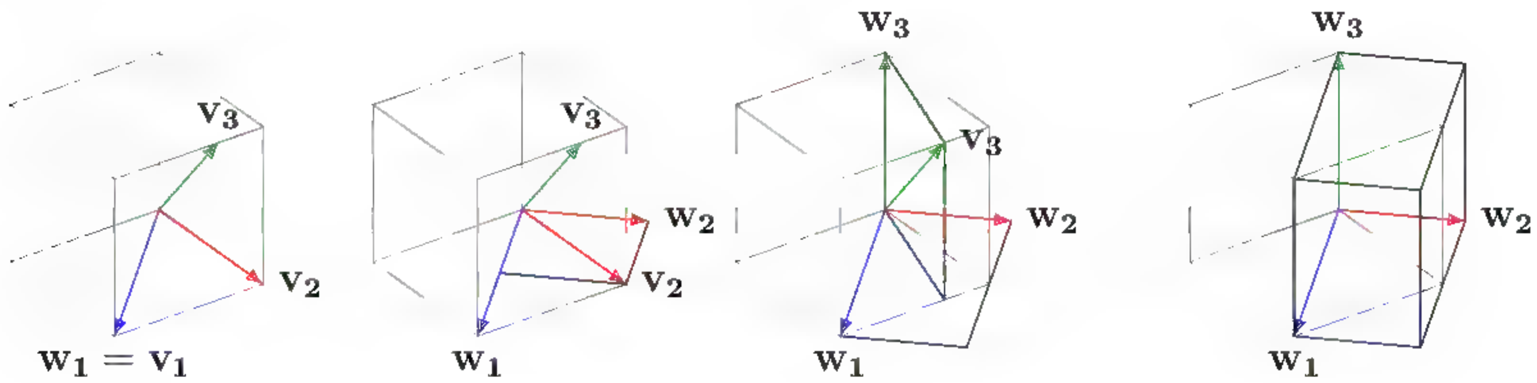


FIGURE 19.2.1. Converting a basis $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ to an orthogonal basis $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$.

Gram–Schmidt Process. Let $\mathbf{v}_1, \dots, \mathbf{v}_k$ be nonzero n-vectors with span V in $\mathbf{R}^n$. Define:

$$V_1 = \mathrm{span}(\mathbf{v}_1), V_2 = \mathrm{span}(\mathbf{v}_1, \mathbf{v}_2), V_3 = \mathrm{span}(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3), \dots.$$

The following algorithm gives an orthogonal basis for the span V of all the $\mathbf{v}_j$'s.

- Let $\mathbf{w}_1 = \mathbf{v}_1$ and define $\mathcal{B}_1$ to be $\{\mathbf{w}_1\}$ (an orthogonal basis for V_1!).
- Let $\mathbf{w}_2 = \mathbf{v}_2 - \mathbf{Proj}_{V_1}(\mathbf{v}_2)$ $(= \mathbf{v}_2 - \mathbf{Proj}_{\mathbf{v}_1}(\mathbf{v}_2)$, with $\mathbf{Proj}_{\mathbf{v}_1}(\mathbf{v}_2)$ defined in Proposition 6.1.1). If $\mathbf{w}_2 \neq \mathbf{0}$ then $\mathcal{B}_2 = \{\mathbf{w}_1, \mathbf{w}_2\}$ is an orthogonal basis for V_2, and if $\mathbf{w}_2 = \mathbf{0}$ then $V_2 = V_1$ and define $\mathcal{B}_2$ to be $\mathcal{B}_1$ (so again $\mathcal{B}_2$ is an orthogonal basis for V_2).
- Let $\mathbf{w}_3 = \mathbf{v}_3 - \mathbf{Proj}_{V_2}(\mathbf{v}_3)$ (computing $\mathbf{Proj}_{V_2}$ as in Theorem 6.2.1 using the orthogonal basis $\mathcal{B}_2$ of V_2 found in the previous step!). If $\mathbf{w}_3 \neq \mathbf{0}$ then the collection $\mathcal{B}_3$ consisting of $\mathcal{B}_2$ along with $\mathbf{w}_3$ is an orthogonal basis for V_3, and if $\mathbf{w}_3 = \mathbf{0}$ then $V_3 = V_2$ and we set $\mathcal{B}_3$ to be $\mathcal{B}_2$ (so again $\mathcal{B}_3$ is an orthogonal basis for V_3).
- and so on, defining at the jth step $\mathbf{w}_j = \mathbf{v}_j - \mathbf{Proj}_{V_{j-1}}(\mathbf{v}_j)$ ($\mathbf{Proj}_{V_{j-1}}$ computed as in Theorem 6.2.1 using the orthogonal basis $\mathcal{B}_{j-1}$ found in the $(j-1)$th step!), and $\mathcal{B}_j$ to consist of $\mathcal{B}_{j-1}$ along with $\mathbf{w}_j$ when $\mathbf{w}_j \neq \mathbf{0}$ whereas if $\mathbf{w}_j = \mathbf{0}$ then $V_j = V_{j-1}$ and define $\mathcal{B}_j$ to be $\mathcal{B}_{j-1}$.

After k steps, $\mathcal{B}_k$ consists of the nonzero $\mathbf{w}_i$'s and is an orthogonal basis of V; the corresponding $\mathbf{v}_i$'s are also a basis of V.

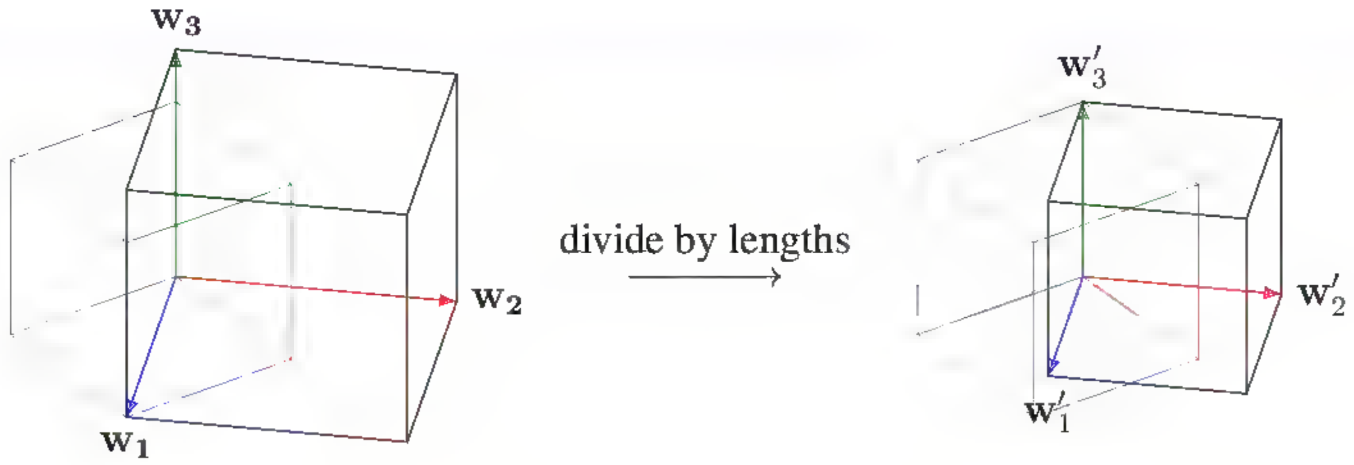


FIGURE 19.2.2. Divide each nonzero $\mathbf{w}_i$ by its length to obtain an orthonormal basis.

Example 19.3.2 gives a case when some $\mathbf{w}_i$ vanishes. If we want an *orthonormal basis* of V, then at the end of the Gram–Schmidt process replace each nonzero $\mathbf{w}_j$ with $\mathbf{w}'_j = \mathbf{w}_j/\|\mathbf{w}_j\|$ (the unit vector pointing in the same direction as $\mathbf{w}_j$), as shown in Figure 19.2.2. When working by hand, it is easier to do this replacement *after* an orthogonal basis is computed, since typically $\|\mathbf{w}_j\|$ is an "ugly" number involving square roots and there is no point in dragging along such a mess for the entire calculation.

This all may look complicated, though we hope that Figure 19.2.1 and Figure 19.2.2 convey the ideas. In Section 19.3 *many* examples are worked out; we strongly encourage keeping the pictures in mind as a guide to what motivates the calculation at each step. We will discuss the relationship between the algebraic and geometric aspects of the Gram–Schmidt[18] process in Remark 19.3.9.

---

[18] As with many results in mathematics, the history behind the naming of the "Gram–Schmidt process" is a bit complicated: the result was actually known (albeit in a less convenient form) long before Gram and Schmidt were born. Gram published the algorithm in 1883, and Schmidt rediscovered it in another context and published it in 1907. Schmidt's paper brought the significance of orthogonalization procedures to widespread attention, and the result has been named after them since around 1935. But the algorithm was already known to and used by Laplace in the period 1812-1820, expressed in a more complicated form that made it difficult to perceive the underlying structure (see [Dry, Sec. 4]).

Applying Theorem 5.2.2 to the output of the Gram–Schmidt process, one obtains the following result (for which a proof is given in Section B.1).

Theorem 19.2.3. If $V = \text{span}(\mathbf{v}_1, \dots, \mathbf{v}_k)$ for k nonzero vectors $\mathbf{v}_1, \dots, \mathbf{v}_k \in \mathbf{R}^n$ then $\dim(V)$ equals the number of nonzero $\mathbf{w}_i$'s obtained from the Gram–Schmidt process. Moreover, the following conditions are equivalent:

(i) $\dim(V) = k$ (equivalently, $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ is a basis of $\text{span}(\mathbf{v}_1, \dots, \mathbf{v}_k)$),
(ii) all $\mathbf{w}_i$'s are nonzero,
(iii) the collection $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ is linearly independent.

In particular, $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ is linearly dependent precisely when some $\mathbf{w}_i$ vanishes, and "basis" is the same as "linearly independent spanning set" for any nonzero linear subspace of $\mathbf{R}^n$.

Remark 19.2.4. Since $\dim \mathbf{R}^n = n$ (see Example 5.3.2), so the only n-dimensional subspace of $\mathbf{R}^n$ is itself (by Theorem 4.2.8), the preceding theorem gives a computable criterion to determine when n given nonzero vectors $\mathbf{v}_1, \dots, \mathbf{v}_n \in \mathbf{R}^n$ are a basis of $\mathbf{R}^n$: it is equivalent that all $\mathbf{w}_i$'s emerging from the Gram–Schmidt process on these n vectors are nonzero. We illustrate this for $n = 3$ in Example 19.3.6.

Since we have seen that *every* nonzero linear subspace V of $\mathbf{R}^n$ has an orthogonal basis (feed a finite spanning set of V into the Gram–Schmidt process and get out an orthogonal basis), we can now explain some geometrically appealing facts concerning orthogonality:

Theorem 19.2.5. If V is a linear subspace of $\mathbf{R}^n$ then the collection $V^\perp$ of n-vectors orthogonal to everything in V (we call $V^\perp$ the *orthogonal complement* of V) is a linear subspace of $\mathbf{R}^n$ and

$$\dim V^\perp = n - \dim V.$$

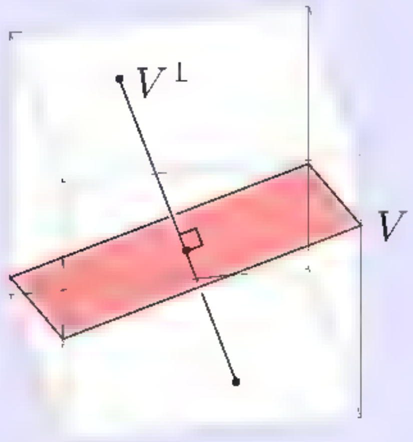


FIGURE 19.2.3. A plane V in $\mathbf{R}^3$ passing through $\mathbf{0}$ and its orthogonal complement line $V^\perp$, with $\dim V^\perp = 1 = 3 - \dim V$.

PROOF. First we handle some special cases. If $V = \{\mathbf{0}\}$ then $V^\perp = \mathbf{R}^n$ and there is nothing to do. If $V = \mathbf{R}^n$ then we are also done, since $(\mathbf{R}^n)^\perp = \{\mathbf{0}\}$ (anything orthogonal to all of the "standard basis vectors" $\mathbf{e}_1, \dots, \mathbf{e}_n$ has each of its coordinates equal to 0 and so must be the zero vector).

Hence, we now may assume that V is a nonzero subspace of $\mathbf{R}^n$ that is not equal to the entirety of $\mathbf{R}^n$, so there is some vector $\mathbf{x} \in \mathbf{R}^n$ not belonging to V. In particular $\mathbf{x}$ cannot be equal to $\mathbf{Proj}_V(\mathbf{x}) \in V$, so the difference $\mathbf{x} - \mathbf{Proj}_V(\mathbf{x}) \in V^\perp$ is nonzero. In other words, now V and $V^\perp$ are both nonzero.

Let $\mathbf{v}_1, \dots, \mathbf{v}_k$ be an orthogonal basis for V (so $k = \dim V$ and $k < n$). Letting $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$ be the standard basis of $\mathbf{R}^n$, the combined collection of nonzero vectors

$$\mathcal{B} = \{\mathbf{v}_1, \dots, \mathbf{v}_k, \mathbf{e}_1, \dots, \mathbf{e}_n\}$$

spans $\mathbf{R}^n$ (as even the last n vectors in the collection do). Now run Gram–Schmidt on this spanning set for $\mathbf{R}^n$. Since the first k vectors in the collection are *already* pairwise orthogonal, the Gram-Schmidt process has no effect on them! So when the Gram–Schmidt process is over, it yields an orthogonal basis of $\mathbf{R}^n$ having the form

$$B = \{\mathbf{v}_1, \dots, \mathbf{v}_k, \mathbf{v}_{k+1}, \dots, \mathbf{v}_n\}$$

for some additional vectors $\mathbf{v}_{k+1}, \dots, \mathbf{v}_n$.

Since the first k vectors in B constitute an orthogonal basis of V, we can give a very explicit description of $V^\perp$: we claim that it is exactly the span of the orthogonal collection of nonzero vectors $\mathbf{v}_{k+1}, \dots, \mathbf{v}_n$. Once this is shown, it would follow that $V^\perp$ is a span of $n - k$ vectors and so $V^\perp$ is a linear subspace moreover with dimension $n - k = n - \dim(V)$ by Theorem 5.2.2 (since we will have exhibited $V^\perp$ as a span of a collection of that many nonzero orthogonal vectors).

So we just have to check that a vector $\mathbf{x} \in \mathbf{R}^n$ is orthogonal to V precisely when it belongs to the span of $\mathbf{v}_{k+1}, \dots, \mathbf{v}_n$. Since B is an orthogonal basis for $\mathbf{R}^n$, we have an expression

$$\mathbf{x} = \sum_{j=1}^{n} c_j \mathbf{v}_j \tag{19.2.2}$$

for some scalars $c_1, \dots, c_n$. Then for any $1 \le i \le n$ we have

$$\mathbf{v}_i \cdot \mathbf{x} = \sum_{j=1}^{n} \mathbf{v}_i \cdot (c_j \mathbf{v}_j) = \sum_{j=1}^{n} c_j (\mathbf{v}_i \cdot \mathbf{v}_j) = c_i (\mathbf{v}_i \cdot \mathbf{v}_i) = c_i \|\mathbf{v}_i\|^2$$

since $\mathbf{v}_i \cdot \mathbf{v}_j = 0$ whenever $j \neq i$. Since $\|\mathbf{v}_i\|^2 \neq 0$ (as $\mathbf{v}_i \neq \mathbf{0}$), we see that $\mathbf{v}_i \cdot \mathbf{x} = 0$ precisely when $c_i = 0$.

By definition $V^\perp$ consists of precisely the vectors $\mathbf{x}$ orthogonal to $\mathbf{v}_1, \dots, \mathbf{v}_k$, which by the preceding calculations is exactly the condition $c_1 = 0, \dots, c_k = 0$. This vanishing says exactly that the first k terms on the right side of (19.2.2) vanish, which is to say

$$\mathbf{x} = c_{k+1}\mathbf{v}_{k+1} + \cdots + c_n \mathbf{v}_n.$$

In other words, we have shown that $V^\perp$ is the span of $\mathbf{v}_{k+1}, \dots, \mathbf{v}_n$, which is what we needed to show. ☐

Example 19.2.6. If L is a line through the origin in $\mathbf{R}^n$, which is to say the span of a nonzero vector $\mathbf{v} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$, then $L^\perp$ consists of those $\mathbf{x} \in \mathbf{R}^n$ satisfying $\mathbf{v} \cdot \mathbf{x} = 0$: this is the solution set to the equation

$$a_1 x_1 + \cdots + a_n x_n = 0$$

with coefficients $a_1, \dots, a_n$ not all equal to 0 (though some might vanish). By Theorem 19.2.5 applied to the 1-dimensional L, we see that $L^\perp$ is a linear subspace with dimension $n - 1$. In fact *every* $(n-1)$-dimensional linear subspace H in $\mathbf{R}^n$ (which we call a *hyperplane*) arises in this way, which is to say that it is the orthogonal complement of a line (exactly in accordance with our visualization for $n = 3$). Indeed, by Theorem 19.2.5 we have that $\dim(H^\perp) = n - \dim(H) = n - (n-1) = 1$, so $L = H^\perp$ is a line through the origin, and hence we just need to affirm that $L^\perp = H$, or in other words $(H^\perp)^\perp = H$. If one thinks about an orthogonal line and plane through the origin in $\mathbf{R}^3$ then such an equality is very plausible.

More generally, if V is *any* linear subspace of $\mathbf{R}^n$ then V is contained in $(V^\perp)^\perp$ (since anything in V is perpendicular to anything in $V^\perp$, by the definition of $V^\perp$) and we claim this is always an equality of linear subspaces (not just when $\dim V = n-1$). Using Theorem 19.2.5 twice, $\dim((V^\perp)^\perp) = n - \dim(V^\perp) = n - (n - \dim V) = \dim V$. Thus, the containment of V in $(V^\perp)^\perp$ involves linear subspaces of $\mathbf{R}^n$ with the *same* dimension, so they must coincide (by Theorem 4.2.8)! ■

19.3. Examples of the Gram–Schmidt process. Our first worked example will transform a collection $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ of 3 nonzero vectors in $\mathbf{R}^4$ into an orthogonal basis $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ of $V = \mathrm{span}(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$. This will be a natural generalization of the method for 2 nonzero vectors in Theorem 7.1.1.

Example 19.3.1. Let V be the span of the 4-vectors

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

We shall construct an orthogonal basis $\{\mathbf{w}_1, \dots\}$ of V.

We begin by defining $\mathbf{w}_1 = \mathbf{v}_1$. Then we subtract from $\mathbf{v}_2$ its projection along $\mathbf{v}_1 = \mathbf{w}_1$: define

$$\mathbf{w}_2 = \mathbf{v}_2 - \mathbf{Proj}_{\mathbf{v}_1} \mathbf{v}_2 = \mathbf{v}_2 - \mathbf{Proj}_{\mathbf{w}_1} \mathbf{v}_2.$$

Explicitly,

$$\begin{aligned} \mathbf{w}_2 = \mathbf{v}_2 - \frac{\mathbf{v}_2 \cdot \mathbf{w}_1}{\mathbf{w}_1 \cdot \mathbf{w}_1} \mathbf{w}_1 = \mathbf{v}_2 - \frac{1}{2}\mathbf{w}_1 &= \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} -1/2 \\ 1/2 \\ 1 \\ 0 \end{bmatrix}. \end{aligned}$$

Theorem 7.1.1 tells us that $\{\mathbf{w}_1, \mathbf{w}_2\}$ is an orthogonal basis of $V_2 = \mathrm{span}(\mathbf{v}_1, \mathbf{v}_2)$. Now comes the key insight from the Gram–Schmidt process: we subtract from $\mathbf{v}_3$ its projection into V_2, which we can compute using the orthogonal basis $\{\mathbf{w}_1, \mathbf{w}_2\}$ of V_2 *that we have already found.* That is, we define:

$$\mathbf{w}_3 = \mathbf{v}_3 - \mathbf{Proj}_{V_2} \mathbf{v}_3 = \mathbf{v}_3 - (\mathbf{Proj}_{\mathbf{w}_1}(\mathbf{v}_3) + \mathbf{Proj}_{\mathbf{w}_2}(\mathbf{v}_3)).$$

The projection of $\mathbf{v}_3$ into V_2 works out as follows:

$$\begin{aligned} \mathbf{Proj}_{V_2} \mathbf{v}_3 &= \mathbf{Proj}_{\mathbf{w}_1} \mathbf{v}_3 + \mathbf{Proj}_{\mathbf{w}_2} \mathbf{v}_3 \\ &= \frac{\mathbf{v}_3 \cdot \mathbf{w}_1}{\mathbf{w}_1 \cdot \mathbf{w}_1} \mathbf{w}_1 + \frac{\mathbf{v}_3 \cdot \mathbf{w}_2}{\mathbf{w}_2 \cdot \mathbf{w}_2} \mathbf{w}_2 \\ &= (1/2)\mathbf{w}_1 + ((1/2)/(3/2))\mathbf{w}_2 \\ &= (1/2)\mathbf{w}_1 + (1/3)\mathbf{w}_2. \end{aligned}$$

Hence,

$$\mathbf{w}_3 = \mathbf{v}_3 - \mathbf{Proj}_{V_2} \mathbf{v}_3 = \mathbf{v}_3 - (1/2)\mathbf{w}_1 - (1/3)\mathbf{w}_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1/2 \\ 1/2 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -1/6 \\ 1/6 \\ 1/3 \\ 0 \end{bmatrix} = \begin{bmatrix} 2/3 \\ -2/3 \\ 2/3 \\ 1 \end{bmatrix}.$$

As a safety check on our work, by computing dot products we can verify directly that $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ is a collection of pairwise orthogonal nonzero vectors (i.e., $\mathbf{w}_1 \cdot \mathbf{w}_2 = 0$, $\mathbf{w}_1 \cdot \mathbf{w}_3 = 0$, $\mathbf{w}_2 \cdot \mathbf{w}_3 = 0$), and they all lie in the span V of the $\mathbf{v}_j$'s by design ($\mathbf{w}_1 = \mathbf{v}_1$ and each $\mathbf{w}_i$ for $i > 1$ is built as a linear combination of $\mathbf{v}_i$ and the previous $\mathbf{w}$'s).

The crucial point is that we can unwind the process and also *express the* $\mathbf{v}_j$*'s as linear combinations of the* $\mathbf{w}_i$*'s.* Indeed, by design we have $\mathbf{v}_1 = \mathbf{w}_1$, $V_2 = \mathrm{span}(\mathbf{v}_1, \mathbf{v}_2) = \mathrm{span}(\mathbf{w}_1, \mathbf{w}_2)$ (so $\mathbf{v}_2$ is a linear combination of $\mathbf{w}_1$ and $\mathbf{w}_2$, or explicitly: since $\mathbf{w}_2 = \mathbf{v}_2 - (1/2)\mathbf{w}_1$, we have $\mathbf{v}_2 = (1/2)\mathbf{w}_1 + \mathbf{w}_2$), and

$$\mathbf{v}_3 = \mathbf{w}_3 + \mathbf{Proj}_{V_2}(\mathbf{v}_3) \in \mathbf{w}_3 + V_2 = \mathbf{w}_3 + \mathrm{span}(\mathbf{w}_1, \mathbf{w}_2),$$

which is contained in $\mathrm{span}(\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3)$ (explicitly: since $\mathbf{w}_3 = \mathbf{v}_3 - (1/2)\mathbf{w}_1 - (1/3)\mathbf{w}_2$, we have $\mathbf{v}_3 = (1/2)\mathbf{w}_1 + (1/3)\mathbf{w}_2 + \mathbf{w}_3$).

Since the $\mathbf{v}_j$'s all belong to the subspace $\mathrm{span}(\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3)$ of $\mathbf{R}^4$, so does everything in the span V of the $\mathbf{v}_j$'s. That is, V is contained in $\mathrm{span}(\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3)$. But we have built every $\mathbf{w}_i$ inside V, so likewise the span of the $\mathbf{w}_i$'s lies inside V. In other words:

$$V = \mathrm{span}(\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3).$$

To summarize, the $\mathbf{w}_i$'s are a collection of pairwise orthogonal nonzero vectors whose span coincides with V, so this collection is an orthogonal *basis* of V (by Theorem 5.2.2). ■

Example 19.3.2. For $\mathbf{v}_1 = \begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix}$, $\mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$, and $\mathbf{v}_3 = \begin{bmatrix} 2 \\ 2 \\ 3 \\ 2 \end{bmatrix}$, define $V = \mathrm{span}(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$. Let's find an orthogonal basis of V.

To solve this, we use the Gram–Schmidt process. The process begins by defining $\mathbf{w}_1 = \mathbf{v}_1 = \begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ and

$$\begin{aligned} \mathbf{w}_2 = \mathbf{v}_2 - \mathbf{Proj}_{\mathbf{w}_1} \mathbf{v}_2 = \mathbf{v}_2 - \frac{\mathbf{w}_1 \cdot \mathbf{v}_2}{\mathbf{w}_1 \cdot \mathbf{w}_1} \mathbf{w}_1 &= \mathbf{v}_2 - \frac{1}{5}\mathbf{w}_1 \\ &= \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \frac{1}{5} \begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} -2/5 \\ 1 \\ 4/5 \\ 1 \end{bmatrix}. \end{aligned}$$

(As a safety-check on the calculation, one can directly compute that $\mathbf{w}_1 \cdot \mathbf{w}_2 = 0$, which is to say that $\mathbf{w}_1$ and $\mathbf{w}_2$ are orthogonal, as they're meant to be.)

Let $V_2 = \mathrm{span}(\mathbf{v}_1, \mathbf{v}_2) = \mathrm{span}(\mathbf{w}_1, \mathbf{w}_2)$, so $\{\mathbf{w}_1, \mathbf{w}_2\}$ is an orthogonal basis of V_2 by design. The next step in the Gram–Schmidt process is to define

$$\mathbf{w}_3 = \mathbf{v}_3 - \mathbf{Proj}_{V_2} \mathbf{v}_3 = \mathbf{v}_3 - \mathbf{Proj}_{\mathbf{w}_1} \mathbf{v}_3 - \mathbf{Proj}_{\mathbf{w}_2} \mathbf{v}_3.$$

We now compute these projections:

$$\mathbf{Proj}_{\mathbf{w}_1}\mathbf{v}_3 = \frac{\mathbf{w}_1\cdot\mathbf{v}_3}{\mathbf{w}_1\cdot\mathbf{w}_1}\mathbf{w}_1 = \frac{7}{5}\mathbf{w}_1 = \frac{7}{5}\begin{bmatrix}2\\0\\1\\0\end{bmatrix} = \begin{bmatrix}14/5\\0\\7/5\\0\end{bmatrix},$$

$$\mathbf{Proj}_{\mathbf{w}_2}\mathbf{v}_3 = \frac{\mathbf{w}_2\cdot\mathbf{v}_3}{\mathbf{w}_2\cdot\mathbf{w}_2}\mathbf{w}_2 = 2\mathbf{w}_2 = 2\begin{bmatrix}-2/5\\1\\4/5\\1\end{bmatrix} = \begin{bmatrix}-4/5\\2\\8/5\\2\end{bmatrix}.$$

Hence,

$$\begin{aligned}\mathbf{w}_3 = \mathbf{v}_3 - \mathbf{Proj}_{\mathbf{w}_1}\mathbf{v}_3 - \mathbf{Proj}_{\mathbf{w}_2}\mathbf{v}_3 &= \mathbf{v}_3 - \frac{7}{5}\mathbf{w}_1 - 2\mathbf{w}_2\\ &= \begin{bmatrix}2\\2\\3\\2\end{bmatrix} - \begin{bmatrix}14/5\\0\\7/5\\0\end{bmatrix} - \begin{bmatrix}-4/5\\2\\8/5\\2\end{bmatrix}\\ &= \begin{bmatrix}0\\0\\0\\0\end{bmatrix}.\end{aligned}$$

Since $\mathbf{w}_3 = \mathbf{0}$, an orthogonal basis for V is given by $\mathbf{w}_1 = \begin{bmatrix}2\\0\\1\\0\end{bmatrix}$ and $\mathbf{w}_2 = \begin{bmatrix}-2/5\\1\\4/5\\1\end{bmatrix}$. In particular, even though V was initially described as the span of 3 vectors $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$, it has as an orthogonal basis $\{\mathbf{w}_1, \mathbf{w}_2\}$ and so V is 2-*dimensional* and the corresponding $\mathbf{v}_i$'s – namely $\{\mathbf{v}_1, \mathbf{v}_2\}$ – constitute a basis.

Geometrically, this means that the original vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ are constrained to lie in a "plane" inside $\mathbf{R}^4$, so according to Theorem 19.2.3 one of the $\mathbf{v}_i$'s is a linear combination of the others. It can be checked by inspection that $\mathbf{v}_3 = \mathbf{v}_1 + 2\mathbf{v}_2$, though finding that relation may seem like magic. This can be found systematically by going back to how the *vanishing* $\mathbf{w}_3$ was built in terms of projections:

$$\mathbf{0} = \mathbf{w}_3 = \mathbf{v}_3 - (7/5)\mathbf{w}_1 - 2\mathbf{w}_2 = \mathbf{v}_3 - (7/5)\mathbf{v}_1 - 2(\mathbf{v}_2 - (1/5)\mathbf{v}_1) = \mathbf{v}_3 - \mathbf{v}_1 - 2\mathbf{v}_2,$$

which again expresses that $\mathbf{v}_3 = \mathbf{v}_1 + 2\mathbf{v}_2$. This makes explicit that $\{\mathbf{v}_1, \mathbf{v}_2\}$ is a basis of V. ■

Remark 19.3.3. If at any stage of the Gram–Schmidt process we obtain a nonzero $\mathbf{w}_j$ which we regard as "ugly" in some way (e.g., it has some fractional entries with big denominators), we can replace $\mathbf{w}_j$ with any nonzero scalar multiple $\mathbf{w}'_j = c\mathbf{w}_j$ we prefer. For instance, in the preceding example we could have replaced $\mathbf{w}_2$ with $\mathbf{w}'_2 = 5\mathbf{w}_2$ that has integer entries (which may simplify work when computing dot products at later steps).

Multiplying some w's by nonzero scalars is harmless because: it doesn't change their span, it doesn't affect whether they are orthogonal to each other, and the projection operations $\mathbf{Proj}_{\mathbf{w}}$ and $\mathbf{Proj}_{\mathbf{w}'}$ are the same for any nonzero $\mathbf{w}$ and $\mathbf{w}' = c\mathbf{w}$ (since $\mathbf{Proj}_{\mathbf{w}}$ only depends on the line spanned by $\mathbf{w}$). In more explicit terms, the agreement of projections is due to factors of c^2 in a numerator and denominator cancelling out:

$$\mathbf{Proj}_{\mathbf{w}'}(\mathbf{v}) = \left(\frac{\mathbf{v}\cdot\mathbf{w}'}{\mathbf{w}'\cdot\mathbf{w}'}\right)\mathbf{w}' = \left(\frac{\mathbf{v}\cdot(c\mathbf{w})}{(c\mathbf{w})\cdot(c\mathbf{w})}\right)c\mathbf{w} = \left(\frac{c(\mathbf{v}\cdot\mathbf{w})}{c^2(\mathbf{w}\cdot\mathbf{w})}\right)c\mathbf{w} = \left(\frac{\mathbf{v}\cdot\mathbf{w}}{\mathbf{w}\cdot\mathbf{w}}\right)\mathbf{w} = \mathbf{Proj}_{\mathbf{w}}(\mathbf{v}).$$

Many references require the Gram–Schmidt process to produce an *orthonormal* basis, which is to say that they divide each nonzero $\mathbf{w}_i$ by its length; please keep this in mind when you encounter the Gram–Schmidt process outside this course. (One merit of orthonormality will arise in Section 22.4.) Our formulation of the Gram–Schmidt process does not require turning each nonzero $\mathbf{w}_i$ into a unit vector by dividing by its length.

There are several reasons that we content ourselves with orthogonality rather than orthonormality. First, for our needs an orthogonal basis is usually sufficient. Second, calculations and formulas would become full of messy square roots if we require turning the nonzero $\mathbf{w}_i$'s into unit vectors during the process, for no real gain since the projection $\mathbf{Proj}_{\mathbf{w}}$ onto the line spanned by a nonzero vector $\mathbf{w}$ is unaffected by replacing $\mathbf{w}$ with $c\mathbf{w}$ for any nonzero scalar c. Finally, if one really needs unit vectors for the (nonzero) output, the division of each nonzero $\mathbf{w}_i$ by its length can always be done *after* the entire process is over rather than along the way (which would force one to drag along annoying square roots).

In homework and especially exams, we design all calculations to have the $\mathbf{w}_j$'s work out to have integer entries or at worst a small single-digit denominator, so there is no need for the scaling trick as in Remark 19.3.3. But just to illustrate how it can be applied in a setting that isn't messy, here is such an example:

Example 19.3.4. For the 4-vectors $\mathbf{v}_1 = \begin{bmatrix}1\\1\\0\\0\end{bmatrix}$, $\mathbf{v}_2 = \begin{bmatrix}0\\1\\1\\0\end{bmatrix}$, $\mathbf{v}_3 = \begin{bmatrix}1\\1\\1\\1\end{bmatrix}$, let's compute an orthogonal basis for their span V in $\mathbf{R}^4$ (in particular finding $\dim V$) and determine if the $\mathbf{v}_i$'s are linearly independent.

Define $\mathbf{w}_1 = \mathbf{v}_1 = \begin{bmatrix}1\\1\\0\\0\end{bmatrix}$, and then

$$\mathbf{w}_2 = \mathbf{v}_2 - \frac{\mathbf{v}_2 \cdot \mathbf{w}_1}{\mathbf{w}_1 \cdot \mathbf{w}_1}\mathbf{w}_1 = \mathbf{v}_2 - \frac{1}{2}\mathbf{w}_1 = \begin{bmatrix}0\\1\\1\\0\end{bmatrix} - \frac{1}{2}\begin{bmatrix}1\\1\\0\\0\end{bmatrix} = \begin{bmatrix}-1/2\\1/2\\1\\0\end{bmatrix}.$$

There is no harm is replacing this with any nonzero scalar multiple, so to get rid of the fractions let's multiply it by 2: we define

$$\mathbf{w}_2' = 2\mathbf{w}_2 = 2\begin{bmatrix}-1/2\\1/2\\1\\0\end{bmatrix} = \begin{bmatrix}-1\\1\\2\\0\end{bmatrix}.$$

Finally, we calculate

$$\mathbf{w}_3 = \mathbf{v}_3 - \frac{\mathbf{v}_3 \cdot \mathbf{w}_1}{\mathbf{w}_1 \cdot \mathbf{w}_1}\mathbf{w}_1 - \frac{\mathbf{v}_3 \cdot \mathbf{w}_2'}{\mathbf{w}_2' \cdot \mathbf{w}_2'}\mathbf{w}_2' = \mathbf{v}_3 - \frac{2}{2}\mathbf{w}_1 - \frac{2}{6}\mathbf{w}_2' = \mathbf{v}_3 - \mathbf{w}_1 - \frac{1}{3}\mathbf{w}_2',$$

and substituting in the explicit vectors $\mathbf{v}_3, \mathbf{w}_1, \mathbf{w}_2'$ turns this into

$$\mathbf{w}_3 = \begin{bmatrix}1\\1\\1\\1\end{bmatrix} - \begin{bmatrix}1\\1\\0\\0\end{bmatrix} - \frac{1}{3}\begin{bmatrix}-1\\1\\2\\0\end{bmatrix} = \begin{bmatrix}0\\0\\1\\1\end{bmatrix} - \begin{bmatrix}-1/3\\1/3\\2/3\\0\end{bmatrix} = \begin{bmatrix}1/3\\-1/3\\1/3\\1\end{bmatrix}.$$

Let's multiply through by 3 to work with $\mathbf{w}_3' = 3\mathbf{w}_3 = \begin{bmatrix} 1 \\ -1 \\ 1 \\ 3 \end{bmatrix}$ instead. Thus, V has as an orthogonal basis $\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \\ 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \\ 1 \\ 3 \end{bmatrix}$. (One should always do the safety check of computing dot products among these to make sure they really are orthogonal, as works out in this case.) There are three such vectors (i.e., no nonzero $\mathbf{w}$'s), so $\dim V = 3$ and the original three vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ are linearly independent. ■

Next we give an example in which we get a vanishing $\mathbf{w}_i$ in the middle of the process, rather than at the end as in Example 19.3.2.

Example 19.3.5. For the 4-vectors $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{v}_3 = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \end{bmatrix}, \mathbf{v}_4 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$, let's compute an orthogonal basis for their span V in $\mathbf{R}^4$ (in particular, finding $\dim V$) and determine if the collection of $\mathbf{v}_i$'s is linearly independent.

Define $\mathbf{w}_1 = \mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$, and then

$$\mathbf{w}_2 = \mathbf{v}_2 - \frac{\mathbf{v}_2 \cdot \mathbf{w}_1}{\mathbf{w}_1 \cdot \mathbf{w}_1}\mathbf{w}_1 = \mathbf{v}_2 - \frac{2}{4}\mathbf{w}_1 = \mathbf{v}_2 - \frac{1}{2}\mathbf{w}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 1/2 \\ -1/2 \\ -1/2 \end{bmatrix}.$$

To avoid fractional entries, we'll work with $\mathbf{w}_2' = 2\mathbf{w}_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$.

Next, we calculate

$$\mathbf{w}_3 = \mathbf{v}_3 - \frac{\mathbf{v}_3 \cdot \mathbf{w}_1}{\mathbf{w}_1 \cdot \mathbf{w}_1}\mathbf{w}_1 - \frac{\mathbf{v}_3 \cdot \mathbf{w}_2'}{\mathbf{w}_2' \cdot \mathbf{w}_2'}\mathbf{w}_2' = \mathbf{v}_3 - \frac{6}{4}\mathbf{w}_1 - \frac{-2}{4}\mathbf{w}_2' = \mathbf{v}_3 - \frac{3}{2}\mathbf{w}_1 + \frac{1}{2}\mathbf{w}_2',$$

and substituting in the explicit vectors $\mathbf{v}_3, \mathbf{w}_1, \mathbf{w}_2'$ turns this into

$$\mathbf{w}_3 = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \end{bmatrix} - \frac{3}{2}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \end{bmatrix} - \begin{bmatrix} 3/2 \\ 3/2 \\ 3/2 \\ 3/2 \end{bmatrix} + \begin{bmatrix} 1/2 \\ 1/2 \\ -1/2 \\ -1/2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Thus $\mathbf{w}_3 = \mathbf{0}$.

Finally, we calculate

$$\begin{aligned}\mathbf{w}_4 = \mathbf{v}_4 - \frac{\mathbf{v}_4\cdot\mathbf{w}_1}{\mathbf{w}_1\cdot\mathbf{w}_1}\mathbf{w}_1 - \frac{\mathbf{v}_4\cdot\mathbf{w}_2'}{\mathbf{w}_2'\cdot\mathbf{w}_2'}\mathbf{w}_2' = \mathbf{v}_4 - \frac{2}{4}\mathbf{w}_1 - \frac{0}{4}\mathbf{w}_2' &= \mathbf{v}_4 - \frac{1}{2}\mathbf{w}_1 \\ &= \begin{bmatrix}1\\0\\0\\1\end{bmatrix} - \frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix} \\ &= \begin{bmatrix}1/2\\-1/2\\-1/2\\1/2\end{bmatrix}.\end{aligned}$$

(There is no $\mathbf{w}_3$-term since $\mathbf{w}_3 = \mathbf{0}$.) Once again, to avoid fractions we'll instead use $\mathbf{w}_4' = 2\mathbf{w}_4 = \begin{bmatrix}1\\-1\\-1\\1\end{bmatrix}$.

Thus, V has as an orthogonal basis

$$\{\mathbf{w}_1, \mathbf{w}_2', \mathbf{w}_4'\} = \left\{\begin{bmatrix}1\\1\\1\\1\end{bmatrix}, \begin{bmatrix}1\\1\\-1\\-1\end{bmatrix}, \begin{bmatrix}1\\-1\\-1\\1\end{bmatrix}\right\}.$$

We conclude that $\dim V = 3$, with the corresponding $\mathbf{v}_i$'s (namely: $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4$) as a basis too, and the original four 4-vectors $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, $\mathbf{v}_4$ must be linearly dependent. More specifically, the vanishing of $\mathbf{w}_3$ encodes a dependence relation among $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, and we can even find it from the vanishing of $\mathbf{w}_3$:

$$\begin{aligned}\mathbf{0} = \mathbf{w}_3 = \mathbf{v}_3 - \frac{3}{2}\mathbf{w}_1 + \frac{1}{2}\mathbf{w}_2' = \mathbf{v}_3 - \frac{3}{2}\mathbf{v}_1 + \frac{1}{2}(2\mathbf{w}_2) &= \mathbf{v}_3 - \frac{3}{2}\mathbf{v}_1 + \mathbf{w}_2 \\ &= \mathbf{v}_3 - \frac{3}{2}\mathbf{v}_1 + (\mathbf{v}_2 - \frac{1}{2}\mathbf{w}_1) \\ &= \mathbf{v}_3 - \frac{3}{2}\mathbf{v}_1 + (\mathbf{v}_2 - \frac{1}{2}\mathbf{v}_1) \\ &= \mathbf{v}_3 - 2\mathbf{v}_1 + \mathbf{v}_2.\end{aligned}$$

■

In Examples 19.3.2 we saw a triple of n-vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ for which one of the $\mathbf{w}_i$'s vanishes (and the $\mathbf{v}_j$'s span a plane). In the next couple of examples of three n-vectors, such vanishing doesn't occur (this is the typical scenario: a random collection of three n-vectors is unlikely to live in a "plane" when $n > 2$).

Example 19.3.6. Consider the span V of the three nonzero 3-vectors

$$\mathbf{v}_1 = \begin{bmatrix}1\\0\\2\end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix}-1\\1\\-1\end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix}-2\\1\\2\end{bmatrix}. \tag{19.3.1}$$

The collection of $\mathbf{v}_i$'s is far from orthogonal: all three dot products $\mathbf{v}_1\cdot\mathbf{v}_2$, $\mathbf{v}_1\cdot\mathbf{v}_3$, and $\mathbf{v}_2\cdot\mathbf{v}_3$ are nonzero. Running the Gram–Schmidt process on this triple will yield an orthogonal basis for their span V.

As usual, we begin with $\mathbf{w}_1 = \mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}$, so to compute

$$\mathbf{w}_2 = \mathbf{v}_2 - \mathbf{Proj}_{\mathbf{w}_1}(\mathbf{v}_2) = \mathbf{v}_2 - \left(\frac{\mathbf{v}_2 \cdot \mathbf{w}_1}{\mathbf{w}_1 \cdot \mathbf{w}_1}\right) \mathbf{w}_1$$

we first evaluate some dot products:

$$\mathbf{w}_1 \cdot \mathbf{w}_1 = 5, \ \ \mathbf{v}_2 \cdot \mathbf{w}_1 = -1 - 2 = -3.$$

Hence,

$$\mathbf{w}_2 = \mathbf{v}_2 - \frac{-3}{5}\mathbf{w}_1 = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 3/5 \\ 0 \\ 6/5 \end{bmatrix} = \begin{bmatrix} -2/5 \\ 1 \\ 1/5 \end{bmatrix} = \frac{1}{5}\begin{bmatrix} -2 \\ 5 \\ 1 \end{bmatrix}.$$

As a safety check on the arithmetic, one can directly verify the orthogonality $\mathbf{w}_2 \cdot \mathbf{w}_1 = 0$.

By the Gram–Schmidt process $\{\mathbf{w}_1, \mathbf{w}_2\}$ is an orthogonal basis for $V_2 = \mathrm{span}(\mathbf{v}_1, \mathbf{v}_2)$, so we can use it to compute $\mathbf{Proj}_{V_2}$ (as we shall need at the next stage):

$$\mathbf{w}_3 = \mathbf{v}_3 - \mathbf{Proj}_{V_2}(\mathbf{v}_3) = \mathbf{v}_3 - (\mathbf{Proj}_{\mathbf{w}_1}(\mathbf{v}_3) + \mathbf{Proj}_{\mathbf{w}_2}(\mathbf{v}_3)).$$

To evaluate

$$\mathbf{Proj}_{\mathbf{w}_1}(\mathbf{v}_3) = \left(\frac{\mathbf{v}_3 \cdot \mathbf{w}_1}{\mathbf{w}_1 \cdot \mathbf{w}_1}\right) \mathbf{w}_1, \ \ \mathbf{Proj}_{\mathbf{w}_2}(\mathbf{v}_3) = \left(\frac{\mathbf{v}_3 \cdot \mathbf{w}_2}{\mathbf{w}_2 \cdot \mathbf{w}_2}\right) \mathbf{w}_2$$

we need to compute some dot products:

$$\mathbf{w}_1 \cdot \mathbf{w}_1 = 5, \ \ \mathbf{v}_3 \cdot \mathbf{w}_1 = 2, \ \ \mathbf{w}_2 \cdot \mathbf{w}_2 = \frac{30}{25}, \ \ \mathbf{v}_3 \cdot \mathbf{w}_2 = \frac{11}{5}.$$

Hence,

$$\mathbf{w}_3 = \mathbf{v}_3 - \frac{2}{5}\mathbf{w}_1 - \frac{11/5}{30/25}\mathbf{w}_2 = \mathbf{v}_3 - \frac{2}{5}\mathbf{w}_1 - \frac{11}{6}\mathbf{w}_2.$$

Plugging in the numbers for $\mathbf{v}_3, \mathbf{w}_1, \mathbf{w}_2$, this becomes

$$\mathbf{w}_3 = \begin{bmatrix} -2 \\ 1 \\ 2 \end{bmatrix} - \begin{bmatrix} 2/5 \\ 0 \\ 4/5 \end{bmatrix} - \frac{11}{30}\begin{bmatrix} -2 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} -5/3 \\ -5/6 \\ 5/6 \end{bmatrix}$$

(where the final equality involves some mechanical steps of adding fractions in each entry). As a safety check on the arithmetic, one can directly verify the orthogonality $\mathbf{w}_3 \cdot \mathbf{w}_1 = 0$ and $\mathbf{w}_3 \cdot \mathbf{w}_2 = 0$.

All of the $\mathbf{w}_j$'s are nonzero, so $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$ is an orthogonal basis for V. Since there are three such $\mathbf{w}_j$'s, we conclude that $\dim(V) = 3$, so the spanning triple $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ for V must be a basis (and so must be linearly independent). In particular, $V = \mathbf{R}^3$ by Remark 19.2.4, so the three vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ actually span $\mathbf{R}^3$. Note that this spanning has been established by geometric reasons, so for each $\mathbf{v} \in \mathbf{R}^3$ we did not need to explicitly find scalars $c_1, c_2, c_3 \in \mathbf{R}$ for which $\mathbf{v} = \sum_{i-1}^{3} c_i \mathbf{v}_i$. This demonstrates the power of using geometry alongside algebra.

Of course, if we *do* want to find those coefficients c_1, c_2, c_3, the way to do it is first to expand $\mathbf{v}$ in terms of the orthogonal basis of $\mathbf{w}_j$'s using the Fourier formula

$$\mathbf{v} = \sum_{j-1}^{3} \left(\frac{\mathbf{v} \cdot \mathbf{w}_j}{\mathbf{w}_j \cdot \mathbf{w}_j}\right) \mathbf{w}_j$$

from Theorem 5.3.6 and then plug in the expression for each $\mathbf{w}_j$ in terms of the $\mathbf{v}_i$'s by unraveling the stages of the Gram–Schmidt process. This was carried out in an $\mathbf{R}^4$-setting in Example 5.3.8. To do the same here, after applying the Fourier formula (Theorem 5.3.6) to express a given $\mathbf{v} \in \mathbf{R}^3$ as a linear combination of the $\mathbf{w}_j$'s we get expressions for the $\mathbf{w}_j$'s in terms of the $\mathbf{v}_i$'s by going back to the Gram–Schmidt process:

$$\mathbf{w}_1 = \mathbf{v}_1, \quad \mathbf{w}_2 = \mathbf{v}_2 + \frac{3}{5}\mathbf{w}_1 = \mathbf{v}_2 + \frac{3}{5}\mathbf{v}_1,$$

and

$$\begin{aligned}\mathbf{w}_3 = \mathbf{v}_3 - \frac{2}{5}\mathbf{w}_1 - \frac{11}{6}\mathbf{w}_2 &= \mathbf{v}_3 - \frac{2}{5}\mathbf{v}_1 - \frac{11}{6}(\mathbf{v}_2 + \frac{3}{5}\mathbf{v}_1)\\ &= \mathbf{v}_3 - \frac{11}{6}\mathbf{v}_2 - \left(\frac{2}{5} + \frac{33}{30}\right)\mathbf{v}_1\\ &= \mathbf{v}_3 - \frac{11}{6}\mathbf{v}_2 - \frac{3}{2}\mathbf{v}_1.\end{aligned}$$

As an illustration, if we choose $\mathbf{v} = \begin{bmatrix}1\\2\\3\end{bmatrix}$ then how can we write $\mathbf{v}$ as a linear combination of $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$? First in terms of $\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{w}_3$,

$$\mathbf{v} = \sum_{j=1}^{3}\left(\frac{\mathbf{v}\cdot\mathbf{w}_j}{\mathbf{w}_j\cdot\mathbf{w}_j}\right)\mathbf{w}_j = \frac{7}{5}\mathbf{w}_1 + \frac{11/5}{6/5}\mathbf{w}_2 + \frac{-5/6}{25/6}\mathbf{w}_3 = \frac{7}{5}\mathbf{w}_1 + \frac{11}{6}\mathbf{w}_2 - \frac{1}{5}\mathbf{w}_3.$$

Plugging in the above expressions for the $\mathbf{w}_j$'s in terms of the $\mathbf{v}_i$'s yields

$$\begin{aligned}\begin{bmatrix}1\\2\\3\end{bmatrix} &= \frac{7}{5}\mathbf{v}_1 + \frac{11}{6}(\mathbf{v}_2 + \frac{3}{5}\mathbf{v}_1) - \frac{1}{5}(\mathbf{v}_3 - \frac{11}{6}\mathbf{v}_2 - \frac{3}{2}\mathbf{v}_1)\\ &= \left(\frac{7}{5} + \frac{11}{10} + \frac{3}{10}\right)\mathbf{v}_1 + \left(\frac{11}{6} + \frac{11}{30}\right)\mathbf{v}_2 - \frac{1}{5}\mathbf{v}_3\\ &= \frac{14}{5}\mathbf{v}_1 + \frac{11}{5}\mathbf{v}_2 - \frac{1}{5}\mathbf{v}_3.\end{aligned}$$

As a safety check, plugging (19.3.1) into this final linear combination does indeed yield $\begin{bmatrix}1\\2\\3\end{bmatrix}$ (check!). ■

Example 19.3.7. We now revisit the linear subspace W of $\mathbf{R}^5$ in Example 4.1.13, for which we found an explicit spanning set consisting of the following three vectors:

$$\mathbf{v}_1 = \begin{bmatrix}-1\\1\\-2/3\\0\\0\end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix}1\\0\\-1/3\\1\\0\end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix}-2\\0\\1/3\\0\\1\end{bmatrix}.$$

We verified their linear independence by a short calculation in Example 19.1.7 that relies on the special arrangement of entries equal to 0 in this vector. Now we shall use the Gram–Schmidt process (which involves more computational effort than in Example 19.1.7, but has the merit of being a robust method that is applicable without regard to the special placement of 0's as entries in the vectors) to give a different verification that $\dim(W) = 3$.

We have $\mathbf{w}_1 = \mathbf{v}_1 = \begin{bmatrix} -1 \\ 1 \\ -2/3 \\ 0 \\ 0 \end{bmatrix}$, so to compute

$$\mathbf{w}_2 = \mathbf{v}_2 - \mathbf{Proj}_{\mathbf{w}_1}(\mathbf{v}_2) = \mathbf{v}_2 - \left(\frac{\mathbf{v}_2 \cdot \mathbf{w}_1}{\mathbf{w}_1 \cdot \mathbf{w}_1}\right) \mathbf{w}_1$$

we first evaluate the dot products

$$\mathbf{w}_1 \cdot \mathbf{w}_1 = \frac{22}{9}, \quad \mathbf{v}_2 \cdot \mathbf{w}_1 = \frac{-7}{9}.$$

Thus,

$$\mathbf{w}_2 = \mathbf{v}_2 - \frac{-7/9}{22/9}\mathbf{w}_1 = \begin{bmatrix} 1 \\ 0 \\ -1/3 \\ 1 \\ 0 \end{bmatrix} + \frac{7}{22}\begin{bmatrix} -1 \\ 1 \\ -2/3 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 15/22 \\ 7/22 \\ -6/11 \\ 1 \\ 0 \end{bmatrix} = \frac{1}{22}\begin{bmatrix} 15 \\ 7 \\ -12 \\ 22 \\ 0 \end{bmatrix}.$$

(One can directly verify $\mathbf{w}_1 \cdot \mathbf{w}_2 = 0$ as a safety check; for this purpose the scalar factor $1/22$ can be ignored.)

The pair of vectors $\{\mathbf{w}_1, \mathbf{w}_2\}$ is an orthogonal basis of $V_2 = \text{span}(\mathbf{v}_1, \mathbf{v}_2)$, so

$$\mathbf{w}_3 = \mathbf{v}_3 - \mathbf{Proj}_{V_2}(\mathbf{v}_3) = \mathbf{v}_3 - (\mathbf{Proj}_{\mathbf{w}_1}(\mathbf{v}_3) + \mathbf{Proj}_{\mathbf{w}_2}(\mathbf{v}_3)) = \mathbf{v}_3 - \left(\frac{\mathbf{v}_3 \cdot \mathbf{w}_1}{\mathbf{w}_1 \cdot \mathbf{w}_1}\right) \mathbf{w}_1 - \left(\frac{\mathbf{v}_3 \cdot \mathbf{w}_2}{\mathbf{w}_2 \cdot \mathbf{w}_2}\right) \mathbf{w}_2$$

with the dot product evaluations

$$\mathbf{w}_1 \cdot \mathbf{w}_1 = \frac{22}{9}, \quad \mathbf{v}_3 \cdot \mathbf{w}_1 = \frac{16}{9}, \quad \mathbf{w}_2 \cdot \mathbf{w}_2 = \frac{41}{22}, \quad \mathbf{v}_3 \cdot \mathbf{w}_2 = \frac{-17}{11},$$

so

$$\begin{aligned}\mathbf{w}_3 = \mathbf{v}_3 - \frac{8}{11}\mathbf{w}_1 + \frac{17/11}{41/22}\mathbf{w}_2 &= \begin{bmatrix} -2 \\ 0 \\ 1/3 \\ 0 \\ 1 \end{bmatrix} - \frac{8}{11}\begin{bmatrix} -1 \\ 1 \\ -2/3 \\ 0 \\ 0 \end{bmatrix} + \frac{17}{451}\begin{bmatrix} 15 \\ 7 \\ -12 \\ 22 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} -29/41 \\ -19/41 \\ 15/41 \\ 34/41 \\ 1 \end{bmatrix}.\end{aligned}$$

(As a safety check on the arithmetic, one can directly verify that $\mathbf{w}_3 \cdot \mathbf{w}_1 = 0$ and $\mathbf{w}_3 \cdot \mathbf{w}_2 = 0$.)

Since $\mathbf{w}_j \neq \mathbf{0}$ for every j, we conclude that $\dim(W) = 3$ (so the spanning set $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ of W is also a basis) by Theorem 19.2.3. ■

Example 19.3.8. Finally, we revisit the linear subspace V of $\mathbf{R}^5$ in Example 5.2.3 that was defined as the span of the three vectors

$$\mathbf{v}_1 = \begin{bmatrix}1\\0\\3\\2\\1\end{bmatrix} \quad \mathbf{v}_2 = \begin{bmatrix}1\\1\\2\\0\\3\end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix}0\\3\\0\\2\\1\end{bmatrix}.$$

The Gram–Schmidt process applied to this triple gives $\mathbf{w}_1 = \mathbf{v}_1 = \begin{bmatrix}1\\0\\3\\2\\1\end{bmatrix}$ and $\mathbf{w}_2 = \mathbf{v}_2 - \mathbf{Proj}_{\mathbf{w}_1}(\mathbf{v}_2) = \mathbf{v}_2 - ((\mathbf{v}_2 \cdot \mathbf{w}_1)/(\mathbf{w}_1 \cdot \mathbf{w}_1))\mathbf{w}_1$ with the dot products

$$\mathbf{w}_1 \cdot \mathbf{w}_1 = 15, \;\; \mathbf{v}_2 \cdot \mathbf{w}_1 = 10.$$

Thus,

$$\mathbf{w}_2 = \mathbf{v}_2 - \frac{10}{15}\mathbf{w}_1 = \begin{bmatrix}1\\1\\2\\0\\3\end{bmatrix} - \frac{2}{3}\begin{bmatrix}1\\0\\3\\2\\1\end{bmatrix} = \begin{bmatrix}1/3\\1\\0\\-4/3\\7/3\end{bmatrix} = \frac{1}{3}\begin{bmatrix}1\\3\\0\\-4\\7\end{bmatrix}.$$

Define $\mathbf{w}_2' = 3\mathbf{w}_2$ to get rid of the denominator. (As a safety check, one directly verifies $\mathbf{w}_1 \cdot \mathbf{w}_2' = 0$.)

The pair of vectors $\{\mathbf{w}_1, \mathbf{w}_2'\}$ is an orthogonal basis of $V_2 = \mathrm{span}(\mathbf{v}_1, \mathbf{v}_2)$, so

$$\mathbf{w}_3 = \mathbf{v}_3 - \mathbf{Proj}_{V_2}(\mathbf{v}_3) = \mathbf{v}_3 - \mathbf{Proj}_{\mathbf{w}_1}(\mathbf{v}_3) - \mathbf{Proj}_{\mathbf{w}_2'}(\mathbf{v}_3),$$

and to compute the projections we evaluate some dot products:

$$\mathbf{v}_3 \cdot \mathbf{w}_1 = 5, \;\; \mathbf{w}_1 \cdot \mathbf{w}_1 = 15, \;\; \mathbf{v}_3 \cdot \mathbf{w}_2' = 8, \;\; \mathbf{w}_2' \cdot \mathbf{w}_2' = 75.$$

Hence, $\mathbf{w}_3 = \mathbf{v}_3 - \frac{5}{15}\mathbf{w}_1 - \frac{8}{75}\mathbf{w}_2' = \mathbf{v}_3 - \frac{1}{3}\mathbf{w}_1 - \frac{8}{75}\mathbf{w}_2'$. Plugging in $\mathbf{v}_3, \mathbf{w}_1, \mathbf{w}_2'$ yields

$$\mathbf{w}_3 = \begin{bmatrix}0\\3\\0\\2\\1\end{bmatrix} - \frac{1}{3}\begin{bmatrix}1\\0\\3\\2\\1\end{bmatrix} - \frac{8}{75}\begin{bmatrix}1\\3\\0\\-4\\7\end{bmatrix} = \frac{1}{25}\begin{bmatrix}-11\\67\\-25\\44\\-2\end{bmatrix}.$$

(It can be verified directly that $\mathbf{w}_3 \cdot \mathbf{w}_1 = 0$ and $\mathbf{w}_3 \cdot \mathbf{w}_2' = 0$, as a safety check.)

As in Example 19.3.7 we have $\mathbf{w}_j \neq 0$ for every j, so $\dim(V) = 3$ and the spanning set $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ of V is a basis by Theorem 19.2.3. ■

Remark 19.3.9. Now that we have worked out many numerical examples of the Gram–Schmidt process, it may be instructive to reflect on how the algebra of the calculations interacts with the geometry of the vectors $\mathbf{v}_i \in \mathbf{R}^n$, at least in the special case that $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ is linearly independent. What information about the $\mathbf{v}_i$'s actually intervenes in the calculation of the orthogonal basis $\{\mathbf{w}_1, \dots, \mathbf{w}_k\}$ of their span?

Nothing happens when we compute $\mathbf{w}_1$, and to compute $\mathbf{w}_2$ we have to compute $\mathbf{Proj}_{\mathbf{w}_1}(\mathbf{v}_2) = \mathbf{Proj}_{\mathbf{v}_1}(\mathbf{v}_2)$, which involves $\mathbf{v}_1 \cdot \mathbf{v}_1$ and $\mathbf{v}_2 \cdot \mathbf{v}_1$. The next step involves $\mathbf{Proj}_{\mathbf{w}_1}(\mathbf{v}_3)$ and $\mathbf{Proj}_{\mathbf{w}_2}(\mathbf{v}_3)$, so in effect it requires computing the dot products

$$\mathbf{w}_1 \cdot \mathbf{w}_1, \;\; \mathbf{w}_1 \cdot \mathbf{v}_3, \;\; \mathbf{w}_2 \cdot \mathbf{w}_2, \;\; \mathbf{w}_2 \cdot \mathbf{v}_3. \tag{19.3.2}$$

By design $\mathbf{w}_1$ and $\mathbf{w}_2$ are built as explicit linear combinations of $\mathbf{v}_1$ and $\mathbf{v}_2$, so by the properties of dot products as in Theorem 2.2.1(iii) we see that to compute the dot products in (19.3.2) we just need to know dot products $\mathbf{v}_i \cdot \mathbf{v}_j$ for $1 \le i, j \le 3$.

Continuing in this way, the only information about $\mathbf{v}_1, \dots, \mathbf{v}_k$ that enters into the Gram–Schmidt process is knowledge of the dot products $\mathbf{v}_i \cdot \mathbf{v}_j$. But taking $i = j$ amounts to knowledge of each $\mathbf{v}_i \cdot \mathbf{v}_i = \|\mathbf{v}_i\|^2$ or equivalently the length of each $\mathbf{v}_i$, so then for $i \ne j$ the knowledge of $\mathbf{v}_i \cdot \mathbf{v}_j$ is the same as the knowledge of

$$\frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\|\|\mathbf{v}_j\|} = \cos(\theta_{ij})$$

for the angle θ_{ij} between $\mathbf{v}_i$ and $\mathbf{v}_j$. Hence, we are really using a *tremendous* amount of geometric information: the lengths of and angles between the $\mathbf{v}_i$'s.

If one knows the values of those dot products then one can (in principle) describe the process of computing an orthogonal basis for the span in purely algebraic terms. This would lose the geometric insight that *motivates* the process, so it would be rather less illuminating. Nonetheless, the actual mechanics of the computation are pure algebra with the values of those dot products.

In the mathematical model for modern portfolio pricing mentioned in Example 4.1.9 with n assets, one is in the above setting with $k = n - 2$ and a linearly independent collection $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ in $\mathbf{R}^n$ about which nothing is known (these are "random vectors") *except* for the values of the dot products $\mathbf{v}_i \cdot \mathbf{v}_j$ (which are fixed by economic data). Those dot products are all that one needs to find an orthogonal basis $\{\mathbf{w}_1, \dots, \mathbf{w}_k\}$ for the span V of the $\mathbf{v}_i$'s, and hence to compute the projection operation

$$\mathbf{Proj}_V = \mathbf{Proj}_{\mathbf{w}_1} + \cdots + \mathbf{Proj}_{\mathbf{w}_k}. \tag{19.3.3}$$

This computation of $\mathbf{Proj}_V$ on a specific n-vector is the entire content of the formula to solve the modern portfolio pricing problem (as least in its most basic formulation). Hence, Gram–Schmidt imbues that formula with geometric meaning (and even leads to a short derivation of it based on (19.3.3) and a 1-line computation in probability theory [KK, Thm. 11(1), Thm. 10]).

19.4. Lagrange multipliers with multiple constraints. An interesting context for the notion of linear independence is in the formulation of a version of Theorem 12.2.1 on Lagrange multipliers for the case of multiple constraints. To formulate this, for a function $f : \mathbf{R}^n \to \mathbf{R}$ and r additional functions $g_1, \dots, g_r : \mathbf{R}^n \to \mathbf{R}$ consider the problem of finding a maximum or minimum for $f(\mathbf{x})$ subject to the simultaneous constraints $g_1(\mathbf{x}) = c_1, \dots, g_r(\mathbf{x}) = c_r$ for some scalars $c_1, \dots, c_r$. For $r = 1$, this is the setup for Theorem 12.2.1. (The case where some of the constraints are inequalities is a rather different type of problem, so we don't discuss that here.) The generalization of Theorem 12.2.1 to multiple constraints is as follows.

Theorem 19.4.1 (Lagrange multipliers II). Let $\mathbf{a} \in \mathbf{R}^n$ be a maximum (or a minimum) of $f(\mathbf{x})$ subject to the combined constraints $g_i(\mathbf{x}) = c_i$ for all $i = 1, \dots, r$. If the gradients $(\nabla g_1)(\mathbf{a}), \dots, (\nabla g_r)(\mathbf{a}) \in \mathbf{R}^n$ are linearly independent then

$$(\nabla f)(\mathbf{a}) = \sum_{j=1}^{r} \lambda_j (\nabla g_j)(\mathbf{a}) \tag{19.4.1}$$

for some scalars $\lambda_1, \dots, \lambda_r$ (called the "Lagrange multipliers"). In words: $(\nabla f)(\mathbf{a})$ belongs to the span of the $(\nabla g_j)(\mathbf{a})$'s when the latter are linearly independent.

Since this result is often used as a way to *find* such extrema $\mathbf{a}$ (via (19.4.1)), you might think the result is hard to use because we first need to verify the linear independence assumption on the gradients $(\nabla g_i)(\mathbf{a})$ at the point $\mathbf{a}$ that *we don't yet know*. But in practice this is not a problem because usually it turns out that the gradients $(\nabla g_i)(\mathbf{x})$ can be checked to be linearly independent for *all* $\mathbf{x}$ in the constraint region (so one doesn't need to make a special calculation for the point $\mathbf{a}$ that isn't even yet known).

Remark 19.4.2. Let's see that in the case $r = 1$, Theorem 19.4.1 asserts exactly the same thing as Theorem 12.2.1. An equivalent way we could state the result with general r is to omit the linear independence *hypothesis* on the $(\nabla g_i)(\mathbf{a})$'s and move its possible failure into the *conclusion*. That is, we can state the same result in the following way: either $(\nabla f)(\mathbf{a})$ belongs to the span of the $(\nabla g_i)(\mathbf{a})$'s or the $(\nabla g_i)(\mathbf{a})$'s are linearly dependent. When phrased in this equivalent way, for $r = 1$ it becomes literally the formulation of Theorem 12.2.1 because for a vector $\mathbf{v} \in \mathbf{R}^n$ the condition that it vanishes is exactly the condition that the collection consisting just of that single vector is linearly dependent (by definition of "linear dependence" for a single vector; see Definition 19.1.1).

Example 19.4.3. An important application of Lagrange multipliers with three constraints ($r = 3$ in Theorem 19.4.1) arises in statistical physics (a subject largely initiated by J.W. Gibbs and Ludwig Boltzmann in the second half of the 19th century, after some initial work by J.C. Maxwell): determine in terms of energy the most likely distribution of arrangements of atoms in a macroscopic sample of matter at a fixed temperature and pressure. One of the Lagrange multipliers will cancel out in the analysis below, and the other two multipliers will encode information about temperature and pressure.

Consider a sample of matter made up of N smaller portions, with N large. The smaller portions each have the same size and contain the same types and numbers of atoms, with the atoms in different microscopic arrangements. Label the possible arrangements of each portion by $j = 1, 2, 3, \dots, r$ and let E_j and V_j denote the respective energy and volume of the jth arrangement. Denote the number of times the jth arrangement of a smaller portion occurs in the large sample as n_j, so

$$\sum_{j=1}^{r} n_j = N. \tag{19.4.2}$$

We are going to regard N as constant and the n_j's as varying subject to the constraint (19.4.2). The total energy E and volume V of the large sample are each a weighted sum of the energies and volumes of the arrangements of the smaller portions:

$$E = \sum_{j=1}^{r} n_j E_j, \quad V = \sum_{j=1}^{r} n_j V_j, \tag{19.4.3}$$

where E, V, and the E_j's and V_j's are constants.

For a given possibility $(n_1, n_2, \dots, n_r)$ of the number of smaller portions of each type, the number of ways that the entire collection of N portions can have n_j of them be of type j is a purely mathematical counting problem: in how many ways can a collection of N things be separated into batches of sizes $n_1, n_2, \dots, n_r$? (For example, if $N = 119$ and $n_1 = 26, n_2 = 37, n_3 = 19, n_4 = 37$ then we are trying to count the number of ways that 119 objects can be separated into 4 batches with respective sizes 26, 37, 19, and 37.) The answer to this problem is given by a formula from combinatorics,

$$\frac{N!}{n_1! n_2! \cdots n_r!}, \tag{19.4.4}$$

where the notation $m!$ (read as "m factorial") for an integer $m = 1, 2, 3, \dots$ denotes the product of all integers from 1 to m (so $1! = 1$, $2! = 1 \cdot 2 = 2$, $3! = 1 \cdot 2 \cdot 3 = 6$, $4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$, $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$, etc.). To determine the most likely distribution of matter is the same as finding

$(n_1, \dots, n_r)$ maximizing $N!/(n_1!n_2! \cdots n_r!)$ subject to the three constraints in (19.4.2) and (19.4.3). The unknown n_j's are varying only through integer values and not a continuous range of options, so this is not strictly a question for calculus (and $m!$ has only been defined for $m = 1, 2, 3, \dots$). But the integers can be arbitrarily large by making the sample bigger and bigger, so they can be regarded as closely spaced relative to their values. The deduction of scientific laws is not a strictly mathematical process, so we'll rewrite our optimization problem in a way that the equations make sense if we pretend that the n_j's are continuous variables and then apply multivariable calculus anyway.

To solve our maximization problem with three constraints, we replace factorials with an expression better suited to calculus. For this purpose, we shall use *Stirling's approximation* $\ln(n!) \approx n\ln(n) - n$ for large n. This approximation for large n is useful because we are interested in the limit that N and the n_j's are all large, corresponding to our macroscopic sample being composed of a very large number of each type of smaller portion (and the use of statistical ideas to explain properties of matter is most appropriate in the limit of large numbers of particles). Because $\ln(x)$ is an increasing function of x, maximizing (19.4.4) is the same as maximizing its logarithm. Using the laws of logarithms and Stirling's approximation, the logarithm of (19.4.4) is

$$\begin{aligned}\ln\left(\frac{N!}{n_1!n_2!\cdots n_r!}\right) = \ln(N!) - \sum_{j=1}^{r}\ln(n_j!) &\approx N\ln(N) - N - \sum_{j=1}^{r}(n_j\ln(n_j) - n_j)\\ &= N\ln(N) - \sum_{j=1}^{r} n_j\ln(n_j),\end{aligned}$$

where the final equality uses that $-N + \sum_{j=1}^{r} n_j = 0$ by (19.4.2).

The upshot is that the most probable distribution of arrangements of atoms in a macroscopic sample of matter corresponds to $(n_1, n_2, \dots, n_r)$ maximizing the function $N\ln(N) - \sum_{j=1}^{r} n_j\ln(n_j)$ subject to the constraints in (19.4.2) and (19.4.3). The factorials have disappeared, and this formulation makes sense mathematically if we pretend the n_j's are positive continuous variables and not merely positive integers! In other words, for $\mathbf{x} = (x_1, \dots, x_r)$ with all $x_i > 0$, we want to maximize $f(\mathbf{x}) = N\ln(N) - \sum_{j=1}^{r} x_j\ln(x_j)$ subject to the constraints that the function $g_1(\mathbf{x}) = \sum_{j=1}^{r} x_j$ is equal to N and the functions $g_2(\mathbf{x}) = \sum_{j=1}^{r} x_jE_j$ and $g_3(\mathbf{x}) = \sum_{j-1}^{r} x_jV_j$ are equal to E and V respectively. We compute the gradients

$$(\nabla g_1)(\mathbf{x}) = \begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}, \quad (\nabla g_2)(\mathbf{x}) = \begin{bmatrix}E_1\\E_2\\\vdots\\E_r\end{bmatrix}, \quad (\nabla g_3)(\mathbf{x}) = \begin{bmatrix}V_1\\V_2\\\vdots\\V_r\end{bmatrix}$$

and in all applications these are linearly independent. Hence, Theorem 19.4.1 tells us that for any a solving the optimization problem there are scalars α, β, γ for which

$$(\nabla f)(\mathbf{a}) = \alpha(\nabla g_1)(\mathbf{a}) + \beta(\nabla g_2)(\mathbf{a}) + \gamma(\nabla g_3)(\mathbf{a}) = \begin{bmatrix}\alpha + \beta E_1 + \gamma V_1\\ \alpha + \beta E_2 + \gamma V_2\\ \vdots\\ \alpha + \beta E_r + \gamma V_r\end{bmatrix},$$

which is to say $\dfrac{\partial f}{\partial x_j}(\mathbf{a}) = \alpha + \beta E_j + \gamma V_j$ for all j. Since the values of f are dimensionless, the multipliers β and γ must be the respective reciprocals of an energy and a volume in order for βE_j and γV_j to be dimensionless (as is needed for $\alpha + \beta E_j + \gamma V_j$ to appear as it does). By inspection

$\partial f/\partial x_j = (x\ln(x))'\big|_{x=x_j} = -1-\ln(x_j)$, so $\ln(a_j) = (-1-\alpha)-\beta E_j-\gamma E_j$ or equivalently $a_j = e^{-1-\alpha}e^{-\beta E_j-\gamma V_j}$. This solves our optimization problem in terms of the Lagrange multipliers α, β, and γ that we don't yet know.

Constraint (19.4.2) yields $N = \sum_{j=1}^{r} a_j = \sum_{j=1}^{r} e^{-1-\alpha}e^{-\beta E_j-\gamma V_j} = e^{-1-\alpha}\sum_{j=1}^{r} e^{-\beta E_j-\gamma V_j}$, so this gives a link between the Lagrange multipliers: if we know β and γ then α is determined. The role of α will soon cancel out, but what can be said about β and γ? Mathematics has served its purpose, and scientific input is now needed. In accordance with the "method of the most probable distribution" in statistical physics, we are primarily interested in the *fraction* of smaller portions that are in the jth type of arrangement of atoms within the most probable distribution. By the preceding work, this fraction is

$$\frac{a_j}{N} = \frac{e^{-1-\alpha}e^{-\beta E_j-\gamma V_j}}{e^{-1-\alpha}\sum_{i=1}^{r} e^{-\beta E_i-\gamma V_i}} = \frac{e^{-\beta(E_j+(\gamma/\beta)V_j)}}{\sum_{i=1}^{r} e^{-\beta(E_i+(\gamma/\beta)V_i)}}.$$

It is shown in thermal physics that β is the reciprocal of an energy proportional to the absolute temperature of the sample: $\beta = 1/(kT)$ for a constant k called Boltzmann's constant (and was introduced by Max Planck). Additional physical arguments show that γ/β is the pressure p of the macroscopic sample of matter (assuming no external forces are acting on it).

We conclude that for the atoms in a macroscopic sample at a given temperature T and pressure p, the frequency of occurrence of the arrangements of type j within the most probable arrangement decreases exponentially with the energy E_j of that arrangement. Thus, the higher the energy of an arrangement of atoms needed for a particular chemical reaction to occur, the exponentially less likely that reaction is to occur in the most probable arrangement at a given temperature and pressure *but* increasing the temperature of the sample will increase the likelihood of that reaction (since $e^{-1/x}$ is increasing in x). This behavior is observed experimentally, and was first explained in 1889 by the Swedish scientist (and winner of the 1903 Nobel Prize in Chemistry) Svante Arrhenius, who worked with Boltzmann during part of the year after earning his PhD.

For the exponent $-\beta(E_j+pV_j)$, as the pressure p is raised we see that pV_j eventually dominates over E_j and so for different j's it is those with smaller volume that become more favorable to occur in the most probable arrangement. In the case of pure carbon at 1 atmosphere of pressure (as encountered at the surface of the earth), a less compact form called graphite (pencil "lead") is more favorable to occur, but above 16,000 atmospheres of pressure the more compact form of diamonds is favored. The force of gravity on the overlying rock makes the pressure rise by around 300 atmospheres for each kilometer of depth below the surface of the earth. Thus, deep enough in the earth diamonds are favored over graphite.

Coming back to the balancing act between energy and pressure in $e^{-(E_j+pV_j)/(kT)}$, when T is room temperature and p is 1 atmosphere the configurations of carbon atoms that must occur for diamond to convert to graphite are too high in energy for such a reaction to happen in any reasonable time. But another option is to make T very big: for any two distinct j and j', the ratio of their frequency of occurrence is

$$\frac{a_j/N}{a_{j'}/N} = \frac{e^{-(E_j+pV_j)/(kT)}}{e^{-(E_{j'}+pV_{j'})/(kT)}} = e^{((E_{j'}-E_j)+p(V_{j'}-V_j))/(kT)} = (e^{1/(kT)})^{(E_{j'}-E_j+p(V_{j'}-V_j))},$$

which becomes closer to 1 as T grows. Hence, for a fixed pressure and sufficiently large temperature we can make any two configurations more equally likely to occur (over a time period of reasonable duration). For example, the conversion of diamond to graphite has been observed at 2800° F on a time scale of hours. For nitroglycerin, the influence of high temperature on its atomic arrangement underlies why a lit fuse causes dynamite to behave as it does, as discovered by Alfred Nobel. ■

Example 19.4.4. A killer app for multiple-constraint Lagrange multipliers is the machine-learning technique called "support vector machines" that we briefly introduced in Example 12.1.2. With the concepts of linear independence and hyperplane ($(n-1)$-dimensional linear subspace of $\mathbf{R}^n$, as discussed in Example 19.2.6) now under our belts, we can describe the problem and its solution. We'll give the most bare-bones version; contemporary applications require many refinements.

Suppose data is given in the form of n-vectors (e.g., genomic sequences, strings of pixel intensities encoding an image, etc.), and we want a computer to learn to classify each data point into one of two types (see Example 12.1.2 for instances of such binary classification tasks). To begin, assume we have a collection of "training data" $\mathbf{t}_1, \dots, \mathbf{t}_r \in \mathbf{R}^n$ whose classification into each type is already known. Using this as input, we want a computer to discover a function of the form

$$\ell(\mathbf{x}) = w_1x_1 + \cdots + w_nx_n - b = \mathbf{w} \cdot \mathbf{x} - b$$

(for some "bias" $b \in \mathbf{R}$ and nonzero vector $\mathbf{w} \in \mathbf{R}^n$ of "weights") so that the two types of data are "most reliably" distinguished by the conditions $\ell(\mathbf{x}) > 0$ and $\ell(\mathbf{x}) < 0$. For $n = 2$, this amounts to a line that "best divides" the training data $\mathbf{t}_1, \dots, \mathbf{t}_r$ into the two types (and the two sides of the line classify new data into one of the two types). How can we find the "best" choice of ℓ; i.e., the "best" b and $\mathbf{w}$?

The condition $\ell(\mathbf{x}) = 0$, or equivalently $\mathbf{w} \cdot \mathbf{x} = b$ (where b may not be 0), is to be visualized as a parallel translation of a hyperplane (it doesn't pass through $\mathbf{0}$ if $b \neq 0$) called an "affine hyperplane" since ℓ is an affine function in the sense of Definition 13.2.1. For $n = 3$ this is the equation of a plane in $\mathbf{R}^3$ that may not pass through $\mathbf{0}$, as studied in Chapter 3. The condition $\ell(\mathbf{x}) = 0$ divides $\mathbf{R}^n$ into two halves, one on each side of this affine hyperplane (for $n = 3$, this is the situation in Example 3.1.2); ℓ is called a *linear classifier*.

Let's call the two classes of data Type I and Type II, and aim to find the "best" function $\ell(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - b$ so that $\ell(\mathbf{x}) > 0$ characterizes Type I and $\ell(\mathbf{x}) < 0$ characterizes Type II. We assume that there are training vectors of each type, and only consider ℓ for which no training vector lies exactly on the affine hyperplane $\ell(\mathbf{x}) = 0$. We also assume that the two types of training vectors really can be separated by an affine hyperplane. (This assumption can fail, with the data perhaps separated by a non-linear equation such as a circle as shown on the left in Figure 19.4.1.

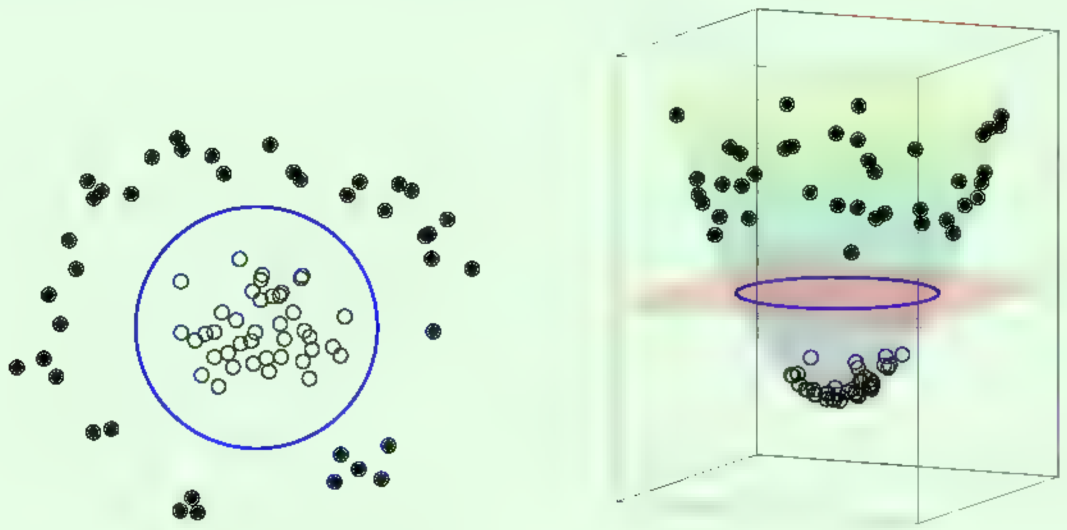


FIGURE 19.4.1. Two types of planar data on the left cannot be separated by a line, but the data can be systematically put into $\mathbf{R}^3$ where it is separated by a plane.

Techniques called "kernel methods" discover non-linear separations by applying linear methods in *higher* dimensions, illustrated by Figure 19.4.1. We will revisit this with more math in Section 24.5.)

Define the sign $s_j = \pm 1$ by the rule $s_j = 1$ when $\mathbf{t}_j$ is of Type I and $s_j = -1$ when $\mathbf{t}_j$ is of Type II. Since $\ell(\mathbf{t}_j) > 0$ when $\mathbf{t}_j$ is of Type I and $\ell(\mathbf{t}_j) < 0$ when $\mathbf{t}_j$ is of Type II, s_j is the sign of $\ell(\mathbf{t}_j)$. Hence, $s_j \ell(\mathbf{t}_j) > 0$ for all j. To *define* the "best" ℓ to fulfill that condition, we want the decision boundary $H = \{\mathbf{x} \in \mathbf{R}^n : \ell(\mathbf{x}) = 0\}$ (an affine hyperplane) to be between but as far as possible from all training vectors: this says the *nearest* of the $\mathbf{t}_j$'s to H should have distance to H as *large* as possible; this is illustrated by the red line in Figure 19.4.2 for the case $n = 2$.

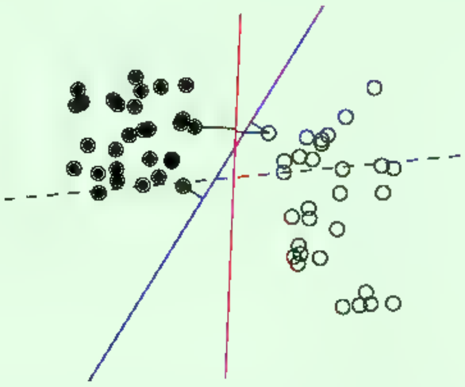


FIGURE 19.4.2. The blue and red lines separate the two types of planar data (the dotted line does not), with data of each type closest to each line indicated by segments orthogonal to that line. The red line is "best": it maximizes the minimal distance.

To find the distance from any $\mathbf{x} \in \mathbf{R}^n$ to H, we use our visual skills from the study of projection in Chapter 6: for the point $\mathbf{p} = \mathbf{Proj}_H(\mathbf{x})$ in H closest to $\mathbf{x}$, the displacement vector $\mathbf{x} - \mathbf{p}$ is along the direction orthogonal to H. Writing an equation for the unknown H in the form $\mathbf{w} \cdot \mathbf{x} = b$ where $\mathbf{w}$ and b need to be found, the direction orthogonal to H is that of the line spanned by $\mathbf{w}$ (for $n = 3$, this is the result in Chapter 3 for obtaining a normal vector to a plane in $\mathbf{R}^3$ from its equation). Hence, as illustrated in Figure 19.4.3, if d is the distance $\|\mathbf{x} - \mathbf{p}\|$ from $\mathbf{x}$ to H then $\mathbf{x} - \mathbf{p}$ is a vector of length d in the line $\mathrm{span}(\mathbf{w})$. In terms of the unit vector $\widehat{\mathbf{w}} = \mathbf{w}/\|\mathbf{w}\|$ in that line, this says $\mathbf{x} - \mathbf{p} = \pm d\widehat{\mathbf{w}}$, where the sign corresponds to the side of H in which $\mathbf{x}$ lies. Note that $\mathbf{w} \cdot \widehat{\mathbf{w}} = (\mathbf{w} \cdot \mathbf{w})/\|\mathbf{w}\| = \|\mathbf{w}\|$.

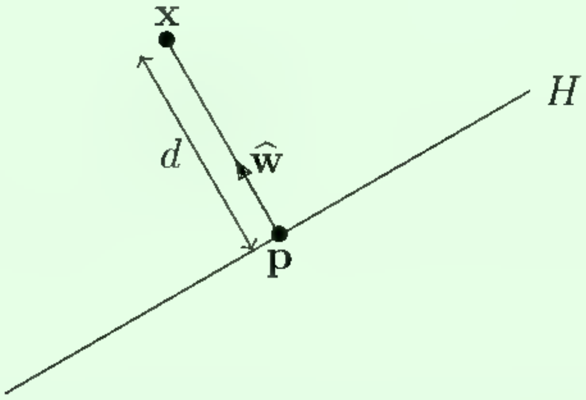


FIGURE 19.4.3. Geometric relationship among $\mathbf{x}$, $\mathbf{p}$, $\mathbf{w}$, and d when the direct path from H to $\mathbf{x}$ points in the same direction as $\mathbf{w}$ (or equivalently, as $\widehat{\mathbf{w}} = \mathbf{w}/\|\mathbf{w}\|$)

For $\mathbf{x} = \mathbf{t}_j$, we conclude $\mathbf{t}_j - \mathbf{p}_j = s_j d_j \widehat{\mathbf{w}}$ with $\mathbf{p}_j \in H$, so $\mathbf{p}_j = \mathbf{t}_j - s_j d_j \widehat{\mathbf{w}}$. By definition $H = \{\mathbf{y} \in \mathbf{R}^n : \ell(\mathbf{y}) = 0\}$, so

$$0 = \ell(\mathbf{p}_j) = \mathbf{w} \cdot \mathbf{p}_j - b = \mathbf{w} \cdot (\mathbf{t}_j - s_j d_j \widehat{\mathbf{w}}) - b = \mathbf{w} \cdot \mathbf{t}_j - s_j d_j (\mathbf{w} \cdot \widehat{\mathbf{w}}) - b = \mathbf{w} \cdot \mathbf{t}_j - s_j d_j \|\mathbf{w}\| - b.$$

Multiplying throughout by $s_j = \pm 1$ and using that $s_j^2 = 1$, solving for d_j gives

$$d_j = s_j(\widehat{\mathbf{w}} \cdot \mathbf{t}_j - b/\|\mathbf{w}\|).$$

Our aim is to *maximize* the minimal value among the distances $d_j > 0$, which is to say we seek $\mathbf{w}$ and b that *maximize* the minimal distance over all r of the training vectors:

$$m = \min_{1 \le j \le r} d_j = \min_{1 \le j \le r} s_j(\widehat{\mathbf{w}} \cdot \mathbf{t}_j - b/\|\mathbf{w}\|) > 0.$$

A positive scaling factor (e.g., $1/\|\mathbf{w}\|$) preserves the direction of inequalities, so it passes through "min": $m = (1/\|\mathbf{w}\|)\min_{1 \le j \le r}(s_j(\mathbf{w} \cdot \mathbf{t}_j) - b)$. Thus, for $m' = m\|\mathbf{w}\| = \min_{1 \le j \le r} s_j(\mathbf{w} \cdot \mathbf{t}_j - b)$, we seek to maximize $m'/\|\mathbf{w}\|$ ($= m$). For any collection of numbers $c_1, \dots, c_r$, their minimum is the *biggest* number M satisfying $c_j \ge M$ for all j. Hence, we seek to maximize $m'/\|\mathbf{w}\|$ subject to the condition

$$s_j(\mathbf{w} \cdot \mathbf{t}_j - b) \ge m' \text{ for every } j.$$

Dividing both sides by $m' > 0$, this is the same as

$$s_j((\mathbf{w}/m') \cdot \mathbf{t}_j - b/m') \ge 1 \text{ for every } j.$$

But $m'/\|\mathbf{w}\| = 1/\|\mathbf{w}/m'\|$, so since it is harmless to scale $\ell(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - b$ by a positive scaling factor (such as $1/m'$) we can rename $\mathbf{w}/m'$ as $\mathbf{w}$ and rename b/m' as b to turn our problem into that of maximizing $1/\|\mathbf{w}\|$ subject to the condition $s_j(\mathbf{w} \cdot \mathbf{t}_j - b) \ge 1$ for every j. Maximizing $1/\|\mathbf{w}\|$ is the same as minimizing $\|\mathbf{w}\|$. That in turn is equivalent to minimizing $\|\mathbf{w}\|^2$, or even $(1/2)\|\mathbf{w}\|^2$. We'll use the latter because the factor $1/2$ will simplify some algebra later.

For any b, define the functions $g_{j,b}(\mathbf{w}) = s_j(\mathbf{w} \cdot \mathbf{t}_j - b) - 1$. We have finally arrived at a good reformulation of our optimization problem, involving many *inequality* constraints:

$$\text{among all } (\mathbf{w}, b) \text{ satisfying } g_{j,b}(\mathbf{w}) \ge 0 \text{ for all } j, \text{ find } (\mathbf{w}^*, b^*) \text{ minimizing } (1/2)\|\mathbf{w}\|^2. \tag{19.4.5}$$

For *all* $\mathbf{t}_j$ of Type I (so $s_j = 1$), "$g_{j,b}(\mathbf{w}) \ge 0$" says "$b \le \mathbf{w} \cdot \mathbf{t}_j - 1$". Hence, $b \le \min_{s_j=1}(\mathbf{w} \cdot \mathbf{t}_j) - 1$. For *all* $\mathbf{t}_j$ of Type II (so $s_j = -1$), "$g_{j,b}(\mathbf{w}) \ge 0$" says "$b \ge \mathbf{w} \cdot \mathbf{t}_j + 1$". Hence, we obtain $b \ge \max_{s_j=-1}(\mathbf{w} \cdot \mathbf{t}_j) + 1$. It can be shown that for the optimum $(\mathbf{w}^*, b^*)$, both inequalities

$$b^* \le \min_{s_j=1}(\mathbf{w}^* \cdot \mathbf{t}_j) - 1, \quad b^* \ge \max_{s_j=-1}(\mathbf{w}^* \cdot \mathbf{t}_j) + 1 \tag{19.4.6}$$

are equalities.

[To prove the inequalities in (19.4.6) are equalities when $(\mathbf{w}^*, b^*)$ is optimal, the first step is to show the affine hyperplane $H = \{\mathbf{w}^* \cdot \mathbf{x} = b^*\}$ solving our problem has the *same* minimal distance from both types of training data. Suppose otherwise, so the least distance to the Type I training data differs from that to the Type II training data. Sliding the hyperplane a small distance parallel to itself (i.e., along its orthogonal line) away from the type of training vectors with smaller minimal distance makes the minimal distance to the *entire* collection of training data slightly larger, contradicting the assumed maximization property of H.

We conclude that among affine hyperplanes $\{\mathbf{x} \in \mathbf{R}^n : \mathbf{w}^* \cdot \mathbf{x} = b^* + c\}$ that are parallel to H, the first ones on each side of H to touch training data have the form $H_\pm = \{\mathbf{w}^* \cdot \mathbf{x} = b^* \pm c\}$ for a *common* $c > 0$, with H_+ touching Type I and H_- touching Type II. The inequalities (19.4.6) respectively say $b^* \le (b^* + c) - 1$ and $b^* \ge (b^* - c) + 1$, which each say $c \ge 1$. If $c > 1$ then $(\mathbf{w}^*/c) \cdot \mathbf{x} - (b^*/c)$ fulfills the constraints yet $\|\mathbf{w}^*/c\| = (1/c)\|\mathbf{w}^*\| < \|\mathbf{w}^*\|$, contradicting the minimization property of $\|\mathbf{w}^*\|$ (or equivalently of $(1/2)\|\mathbf{w}^*\|^2$) under the constraints. This shows $c = 1$, which says (19.4.6) is a pair of equalities.]

For $m_{\mathrm{I}}(\mathbf{w}^*) = \min_{s_j=1}(\mathbf{w}^* \cdot \mathbf{t}_j)$ (minimize over Type I) and $m_{\mathrm{II}}(\mathbf{w}^*) = \max_{s_j=-1}(\mathbf{w}^* \cdot \mathbf{t}_j)$ (maximize over Type II), we have shown $b^* = m_{\mathrm{I}}(\mathbf{w}^*) - 1$ and $b^* = m_{\mathrm{II}}(\mathbf{w}^*) + 1$. Since b^* is equal to both of these expressions, it also equals their average:

$$b^* = \frac{m_{\mathrm{I}}(\mathbf{w}^*) + m_{\mathrm{II}}(\mathbf{w}^*)}{2}. \tag{19.4.7}$$

But what is $\mathbf{w}^*$? A crucial observation is that in practice the number of j's for which $g_{j,b^*}(\mathbf{w}^*) = 0$ is small (compared to n). That is, few training vectors of Type I lie on the hyperplane $\mathbf{w}^* \cdot \mathbf{x} = b^* + 1$ and few of Type II lie on $\mathbf{w}^* \cdot \mathbf{x} = b^* - 1$ (we saw above that for the optimizing affine hyperplane $\mathbf{w}^* \cdot \mathbf{x} = b^*$, its "nearest" parallel translates passing through training data of the respective types are $\mathbf{w}^* \cdot \mathbf{x} = b^* \pm 1$). So for our optimization problem (19.4.5), equality $g_{j,b}(\mathbf{w}) = 0$ holds at the optimum $(\mathbf{w}^*, b^*)$ for very few j's; the corresponding $\mathbf{t}_j$'s are called *support vectors*. We don't yet know which those will be, but it is useful that such equalities hold for few j's.

Now we can solve for $\mathbf{w}^*$ by using the following variant of multiple-constraint Langrange multipliers (Theorem 19.4.1) for the situation of inequality constraints:

Theorem 19.4.5 (Karush-Kuhn-Tucker Conditions, special case). Suppose $\mathbf{a}$ is a local minimum for $f(\mathbf{x})$ subject to conditions $g_1(\mathbf{x}) \geq 0, \ldots, g_r(\mathbf{x}) \geq 0$. For the collection J of indices $1 \leq j \leq r$ with $g_j(\mathbf{a}) = 0$, assume the gradients $(\nabla g_j)(\mathbf{a})$ for $j \in J$ are linearly independent. Then

$$(\nabla f)(\mathbf{a}) = \sum_{j \in J} \lambda_j (\nabla g_j)(\mathbf{a}) \tag{19.4.8}$$

for scalars $\{\lambda_j : j \in J\}$ that are non-negative.

Remark 19.4.6. Since the only j's for which g_j plays a role (i.e., is "active") in the gradient equation (19.4.8) at $\mathbf{a}$ are those for which $g_j(\mathbf{a}) = 0$, the corresponding inequalities $g_j \geq 0$ are called the "active constraints" at $\mathbf{a}$. In applications one often knows, as above, that the number of active constraints at a global minimum is small (compared to the ambient dimension n). Hence, the linear independence hypothesis on the corresponding gradients in Theorem 19.4.5 is usually satisfied there.

Our situation is $f(\mathbf{x}) = (1/2)\|\mathbf{x}\|^2$ and $g_j(\mathbf{x}) = s_j(\mathbf{x} \cdot \mathbf{t}_j - b) - 1$ with b to be determined by (19.4.7) afterwards. Since $(\nabla f)(\mathbf{w}) = \mathbf{w}$ (thanks to the factor $1/2$) and $(\nabla g_j)(\mathbf{w}) = s_j \mathbf{t}_j$, (19.4.8) at $\mathbf{a} = \mathbf{w}^*$ says $\mathbf{w}^* = \sum_{j \in J} \lambda_j s_j \mathbf{t}_j$ for some unknown "multipliers" $\lambda_j \geq 0$ and set of indices J, where moreover $g_j(\mathbf{w}^*) = 0$ for $j \in J$ and $g_j(\mathbf{w}^*) > 0$ for $j \notin J$. The linear independence of the support vectors (in practice) ensures that J and the multipliers λ_j are uniquely determined.

To go further and formulate an algorithm to find the support vectors and the λ_j's, we introduce an important notion in the theory of optimization that arose in a primordial form in the optional Section 12.3 on shadow prices but which we take up from scratch here: consider the "Lagrangian"

$$L(\mathbf{w}, b, \boldsymbol{\lambda}) = f(\mathbf{w}) - \sum_{i=1}^{r} \lambda_i s_i (\mathbf{w} \cdot \mathbf{t}_i - b) \tag{19.4.9}$$

with $\mathbf{w} \in \mathbf{R}^n$, $b \in \mathbf{R}$, and $\boldsymbol{\lambda} \in \mathbf{R}^r$ subject to the conditions $s_i(\mathbf{w} \cdot \mathbf{t}_i - b) \geq 0$ for all i and $\lambda_i \geq 0$ for all i. The term being subtracted in (19.4.9) is always non-negative due to the constraints, so the difference is largest when all λ_i's vanish: $\max_{\boldsymbol{\lambda}} L(\mathbf{w}, b, \boldsymbol{\lambda}) = L(\mathbf{w}, b, \mathbf{0}) = f(\mathbf{w})$. Hence, by varying through all $(\mathbf{w}, b)$ in the constraint region, our problem is the same as "minimizing that maximum".

In the theory of optimization, a result called *Slater's condition* guarantees in our setting that the value of the "minimized maximum" $\min_{(\mathbf{w},b)} \max_{\boldsymbol{\lambda}} L(\mathbf{w}, b, \boldsymbol{\lambda})$ over the entire constraint region of $(\mathbf{w}, b, \boldsymbol{\lambda})$'s is equal to an analogous "maximized minimum" $\max_{\boldsymbol{\lambda}} \min_{(\mathbf{w},b)} L(\mathbf{w}, b, \boldsymbol{\lambda})$. To be precise about the latter expression, first for each $\boldsymbol{\lambda}$ it can be shown using techniques from convex analysis that the minimum $\min_{(\mathbf{w},b)} L(\mathbf{w}, b, \boldsymbol{\lambda})$ is attained at a unique $(\mathbf{w}^*, b^*)$ depending on $\boldsymbol{\lambda}$ and that the gradient of $L(\mathbf{w}, b, \boldsymbol{\lambda})$ as a scalar-valued function of $(\mathbf{w}, b) \in \mathbf{R}^{n+1}$ vanishes at $(\mathbf{w}^*, b^*)$. By staring at the definition of L, the contribution to its vanishing gradient from partials with respect to the w_i's yields $\mathbf{w}^* = \sum_{i=1}^r \lambda_i s_i \mathbf{t}_i$ (similarly to our earlier calculation of $\mathbf{w}^*$ under a linear independence hypothesis) and the contribution from the b-partial yields $\sum_{i=1}^r \lambda_i s_i = 0$. Plugging this expression for $\mathbf{w}^*$ into the definition of the Lagrangian and using the vanishing of $\sum_{i=1}^r \lambda_i s_i$ yields (after some algebraic manipulations):

$$L(\mathbf{w}^*(\boldsymbol{\lambda}), b^*(\boldsymbol{\lambda}), \boldsymbol{\lambda}) = \sum_{i=1}^r \lambda_i - \frac{1}{2} \sum_{i,i'=1}^r s_i s_{i'} (\mathbf{t}_i \cdot \mathbf{t}_{i'}) \lambda_i \lambda_{i'}. \tag{19.4.10}$$

Next, using more convex analysis, the Slater condition and the geometry of the original situation guarantee several things: the "maximized minimum"

$$\max_{\boldsymbol{\lambda}} L(\mathbf{w}^*(\boldsymbol{\lambda}), b^*(\boldsymbol{\lambda}), \boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda}} \min_{(\mathbf{w},b)} L(\mathbf{w}, b, \boldsymbol{\lambda})$$

is attained at some $\boldsymbol{\lambda}^*$, the corresponding $(\mathbf{w}^*, b^*)$ is what we sought above, and $\boldsymbol{\lambda}^*$ is the corresponding collection of non-negative "multipliers" (one per support vector, so that $\boldsymbol{\lambda}^*$ is unique). Hence, rather than focus on the original minimization problem in (19.4.5), we can instead focus on solving the "maximized minimum" problem that is maximizing (19.4.10) under some constraints in $\boldsymbol{\lambda}$. (This change in perspective is an instance of the powerful method called *duality* in linear programming.)

In other words, our task has become to maximize

$$\sum_{i=1}^r \lambda_i - \frac{1}{2} \sum_{i,i'=1}^r s_i s_{i'} (\mathbf{t}_i \cdot \mathbf{t}_{i'}) \lambda_i \lambda_{i'}$$

subject to the constraints $\lambda_1, \ldots, \lambda_r \geq 0$ and $\sum_{i=1}^r \lambda_i s_i = 0$. Observe that the original variables $w_1, \ldots, w_n, b$ have entirely disappeared: we now have an optimization problem solely in terms of the Lagrange multpliers $\lambda_1, \ldots, \lambda_r$; the SMO algorithm solves this problem in practice. The (unique) solution $\boldsymbol{\lambda}$ has $\lambda_i \neq 0$ only for a small collection J of the indices $1 \leq i \leq r$, and in terms of the corresponding support vectors $\mathbf{t}_j$ for $j \in J$ the computer computes $\mathbf{w}^* = \sum_{j \in J} \lambda_j s_j \mathbf{t}_j$ and then computes b^* in terms of $\mathbf{w}^*$ via (19.4.7). Coming back to the original machine learning task, the computer has "learned" a linear classifier: it classifies a new data point $\mathbf{a}$ as being of Type I or Type II according as $\mathbf{w}^* \cdot \mathbf{a} > b^*$ or $\mathbf{w}^* \cdot \mathbf{a} < b^*$. We will revisit this situation with more math in further detail in Section 24.5. ■

Chapter 19 highlights (links to highlights in and next chapters)

Notation	Meaning	Location in text
$V^\perp$	orthogonal complement of a linear subspace V of $\mathbf{R}^n$	Definition 19.2.5

Concept	Meaning	Location in text
linear dependence, linear independence	mathematically precise sense in which a given collection of vectors $\mathbf{v}_1, \dots, \mathbf{v}_k$ in $\mathbf{R}^n$ does or does not have "redundancy" relative to its span	Definition 19.1.1
Gram–Schmidt process	algorithm that constructs an orthogonal basis for a linear subspace of $\mathbf{R}^n$ when given a spanning set of nonzero vectors	boxes before and after Figure 19.2.2
orthogonal complement	for linear subspace V of $\mathbf{R}^n$, it is the collection of all n-vectors orthogonal to everything in V	Theorem 19.2.5

Result	Meaning	Location in text
vectors in a linearly independent collection are nonzero	if $\mathbf{v}_1, \dots, \mathbf{v}_k \in \mathbf{R}^n$ is a linearly independent collection then all $\mathbf{v}_i$'s must be nonzero	Remark 19.1.4
linear independence and dependence via vanishing linear combination	for $\mathbf{v}_1, \dots, \mathbf{v}_k \in \mathbf{R}^n$, they are linearly independent precisely when $c_1\mathbf{v}_1 + \cdots + c_k\mathbf{v}_k = \mathbf{0}$ only for $c_1 = 0, \dots, c_k = 0$ (so linearly dependent precisely when $\sum_{j=1}^k c_j\mathbf{v}_j = \mathbf{0}$ for some scalars $c_1, \dots, c_k$ at least one of which is nonzero)	Theorem 19.1.5
mutually orthogonal nonzero vectors in $\mathbf{R}^n$ are linearly independent	if $\mathbf{w}_1, \dots, \mathbf{w}_k$ in $\mathbf{R}^n$ are nonzero and pairwise orthogonal then $\{\mathbf{w}_1, \dots, \mathbf{w}_k\}$ is a linearly independent collection	Theorem 19.1.5
Gram–Schmidt process yields orthogonal basis and detects linear independence	for nonzero input $\mathbf{v}_1, \dots, \mathbf{v}_k$, the output $\mathbf{w}_1, \dots, \mathbf{w}_k$ has its nonzero members an orthogonal basis for the same span, so $\mathbf{v}_i$'s are linearly independent precisely when *all* $\mathbf{w}_j$'s are nonzero	Theorem 19.2.3
good geometric behavior of orthogonal complement	for linear subspace V of $\mathbf{R}^n$, its orthogonal complement $V^\perp$ is a linear subspace satisfying $\dim V^\perp = n - \dim V$ and $(V^\perp)^\perp = V$	Theorem 19.2.5, Example 19.2.6

Skill	Location in text
relate linear independence to system of linear equations (not solve!)	Examples 19.1.7, 19.1.8
carry out Gram–Schmidt process to make orthogonal (or orthonormal) basis and check if linear independence holds or not	Examples 19.3.1, 19.3.2, 19.3.4–19.3.7
convert vanishing of some $\mathbf{w}_j$ (Gram–Schmidt output) into explicit linear dependence among $\mathbf{v}_i$'s (Gram–Schmidt input)	end of Examples 19.3.2 and 19.3.5

20. Matrix transpose, quadratic forms, and orthogonal matrices

The notion of transpose is a further concept in matrix algebra that, among other things, enables us to work more effectively with dot products. It will also give a way to define and work with a special class of matrices called *symmetric* (that arise in many physical and statistical contexts) and to define and invert a special class of matrices called *orthogonal* that have concrete geometric meaning (essential for all 3D graphics) and arise in many applications (e.g., solving systems of linear equations in Chapter 22, the singular value decomposition in Section 27.3, and the fundamental QR algorithm in Appendix H).

By the end of this chapter, you should be able to:

- transpose a matrix and write the dot product as a 1×1 matrix product involving transposes;
- for a "symmetric" matrix A, directly write out a function of the form $q(\mathbf{v}) = \mathbf{v}^\top A\mathbf{v}$ in terms of the coordinates of $\mathbf{v}$ *without* carrying out the matrix multiplication;
- invert an orthogonal matrix and know that a product of orthogonal matrices is orthogonal.

20.1. The transpose. The *transpose* of a matrix swaps the roles of rows and columns. The transpose of $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ is $\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$. Written out more generally:

Definition 20.1.1. Given an $m \times n$ matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix},$$

its *transpose*, denoted as $A^\top$, is defined to be the $n \times m$ matrix obtained by turning rows into columns and turning columns into rows. In other words, the ith row of $A^\top$ is the ith column of A:

$$A^\top = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix}$$

Equivalently:

$$ij\text{-entry of } A^\top = ji\text{-entry of } A.$$

The transpose can be visualized as "flipping a matrix across its diagonal", so unsurprisingly applying transpose twice returns the original matrix: $(A^\top)^\top = A$. Matrix transpose also respects matrix addition: $(A+B)^\top = A^\top + B^\top$.

Example 20.1.2. Let $A = \begin{bmatrix} 1 & 3 \\ 2 & 5 \end{bmatrix}$ and $B = \begin{bmatrix} 4 & 6 \\ 7 & 1 \end{bmatrix}$, so $A^\top = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix}$ and $B^\top = \begin{bmatrix} 4 & 7 \\ 6 & 1 \end{bmatrix}$. That $(A^\top)^\top = A$, $(B^\top)^\top = B$, and $(A+B)^\top = A^\top + B^\top$ can be checked directly for this specific A and B (please do it) and are similarly verified to hold in general (please try). ■

Why should anyone care about the transpose operation? At first sight, it may look artificial or at least uninteresting. By far the most important reason for interest in the matrix transpose is that it encodes the surprising way that a linear transformation moves "across" a dot product. Before we state the general result, we look at a real-world example where the situation would arise.

Example 20.1.3 (Production matrix). Let's consider raw materials that are each used in the production of two types of cars, and the price per unit of each raw material. Suppose the raw materials are steel, plastics, and aluminum. Let the "quantity vector" $\mathbf{q}$ be the 2-vector whose jth entry q_j is the number of cars being made of the jth type; perhaps $\mathbf{q} = \begin{bmatrix} 200 \\ 50 \end{bmatrix}$ (200 cars of type 1, 50 cars of type 2). Let the "price vector" $\mathbf{p}$ be the 3-vector whose entries are the price in dollars of each unit of material, say p_1 is the price of 1 kg of steel, p_2 is the price of 1 kg of plastics, and p_3 is the price of 1 kg of aluminum.

To compute the total cost of the amount of these raw materials used to make the specified quantities of both final products, we need *more information*: we need to know how much of each raw material is used in the production of each type of car. This will be encoded in a 2×3 matrix: let A be the matrix whose first row is the number of units of each type of raw material to make the first type of car, and whose second row is the number of units of each type of raw material to make the second type of car. For instance, if

$$A = \begin{bmatrix} 750 & 120 & 180 \\ 1200 & 170 & 240 \end{bmatrix}$$

then the first type of car requires 750 kg of steel, 120 kg of plastics, and 180 kg of aluminum, and the second row is the corresponding amount of each raw material required to make a car of the second type.

The total cost mentioned above can be found in two ways. First, the matrix-vector product

$$\mathbf{c} = A\mathbf{p} = \begin{bmatrix} 750p_1 + 120p_2 + 180p_3 \\ 1200p_1 + 170p_2 + 240p_3 \end{bmatrix} \in \mathbf{R}^2$$

has its entries giving the cost of those raw materials used to make a *single* car of each type (do you agree?). If we write $\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$ for this "cost vector" then the dot product $\mathbf{q} \cdot \mathbf{c} = q_1c_1 + q_2c_2 = 200c_1 + 50c_2$ is the total cost (why?). In other words, $\mathbf{q} \cdot (A\mathbf{p})$ is the total amount we sought to find.

But there's another way to proceed: rather than break up the calculation according to each type of car, we could instead focus on the amount of each type of raw material needed to make the specified quantities of both types of cars and *then* take into account the price per unit for each raw material. For this approach, the transpose $A^\top$ is more relevant: the matrix-vector product

$$A^\top \mathbf{q} = \begin{bmatrix} 750q_1 + 1200q_2 \\ 120q_1 + 170q_2 \\ 180q_1 + 240q_2 \end{bmatrix} \in \mathbf{R}^3$$

has as its first entry the total number of units of steel needs to make the entire production amount of cars of both types, and similarly with plastics and aluminum for the second and third entries respectively. This is keeping track of the total raw materials used, so the dot product $(A^\top \mathbf{q}) \cdot \mathbf{p}$ which brings in the price *per unit* of each raw material is also the total amount we sought to find.

Both dot products $(A^\top \mathbf{q}) \cdot \mathbf{p}$ and $\mathbf{q} \cdot (A\mathbf{p})$ compute the same thing, so they are equal to each other. The reason A was used in one approach and $A^\top$ was used in the other is because the total cost can be broken down in two ways: the contribution from each type of car or from each type of raw material. ■

Theorem 20.1.4. For any $m \times n$ matrix A and vectors $\mathbf{x} \in \mathbf{R}^n$ and $\mathbf{y} \in \mathbf{R}^m$,

$$(A\mathbf{x}) \cdot \mathbf{y} = \mathbf{x} \cdot (A^\top \mathbf{y}).$$

In words: a matrix moves across a dot product via its transpose.

In particular, if $A = A^\top$ (called *symmetric* since it says $a_{ij} = a_{ji}$) then $m = n$ (i.e., A is a square matrix) and $(A\mathbf{x}) \cdot \mathbf{y} = \mathbf{x} \cdot (A\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$.

We prove this in Remark 20.1.10 via how transpose and matrix multiplication interact. Theorem 20.1.4 is an algebraic fact with no visual explanation (even though dot products have visual meaning).

Example 20.1.5 (GPT). For two collections $\mathbf{q}_1, \dots, \mathbf{q}_M$ and $\mathbf{k}_1, \dots, \mathbf{k}_N$ of n-vectors ($M \neq N$ is allowed), consider the $M \times n$ matrix Q and $N \times n$ matrix K whose rows are the $\mathbf{q}_i$'s and $\mathbf{k}_j$'s respectively (from top to bottom). The columns of $K^\top$ are the rows $\mathbf{k}_j$ of K, so the entries of $QK^\top$ are exactly the dot products $\mathbf{q}_i \cdot \mathbf{k}_j$ among all of the $\mathbf{q}$'s and $\mathbf{k}$'s. That is the reason such products $QK^\top$ arise in the calculation of attention units within transformer models in machine learning (the "T" in "GPT" stands for "transformer"); this 3Blue1Brown video discusses it further. ■

In Part V we will learn that matrices A that are *symmetric* (i.e., $A^\top = A$; see Definition 20.3.5) satisfy remarkable properties that hold the key to both the multivariable second derivative test and the entirety of modern data analysis (image compression, machine learning, and much more). The essential feature of symmetric matrices is the property "$(A\mathbf{x}) \cdot \mathbf{y} = \mathbf{x} \cdot (A\mathbf{y})$" in the symmetric case of Theorem 20.1.4. It may seem that this should have no relevance for the study of a general $m \times n$ matrix M (such as a big matrix of data, which is essentially never a square matrix, let alone symmetric), but that is incorrect. The link to symmetry is that for any M whatsoever, by Theorem 20.1.4 the squared length $\|M\mathbf{x}\|^2$ is equal to $M\mathbf{x} \cdot M\mathbf{x} = \mathbf{x} \cdot (M^\top M)\mathbf{x}$ and the $n \times n$ product matrix $M^\top M$ is *always* symmetric (see Theorem 20.3.8). The utility of that symmetry rests on a deep theorem about symmetric matrices to be discussed later (Theorem 24.1.4), building on some further concepts introduced in Section 20.3.

Matrices of the form $M^\top M$ ($= NN^\top$ for $N = M^\top$) as in the above discussion are called *Gram* matrices. This class of symmetric matrices arises in: physical situations (see Examples 24.6.7 and 26.1.10), data science (the "singular value decomposition" of M in Section 27.3 expresses features of $M^\top M$), and finance and statistics (if we record n measured features of each of $m > 1$ samples as rows of an $m \times n$ matrix then upon recentering every column – the measurements of each feature – to have average 0 we get an $m \times n$ matrix X, and the $n \times n$ product $(1/(m-1))X^\top X$ is the *covariance matrix* of the data).

Example 20.1.6. For A and B as in Example 20.1.2, we have the Gram matrices (please check):

$$A\,A^\top = \begin{bmatrix} 1 & 3 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix} = \begin{bmatrix} 10 & 17 \\ 17 & 29 \end{bmatrix}, \quad A^\top A = \begin{bmatrix} 5 & 13 \\ 13 & 34 \end{bmatrix}, \quad B\,B^\top = \begin{bmatrix} 52 & 34 \\ 34 & 50 \end{bmatrix}, \quad B^\top B = \begin{bmatrix} 65 & 31 \\ 31 & 37 \end{bmatrix}.$$

So typically for a square matrix M we have $M\,M^\top \neq M^\top M$, but the above examples illustrate that $M\,M^\top$ and $M^\top M$ are symmetric, as mentioned above (a feature we will come back to in Section 20.3). ■

Here are the general properties relating matrix transpose and matrix multiplication, the first of which is reminiscent of what we have seen for matrix inversion (in Examples 18.4.3–18.4.4).

Matrix transpose has the following properties with respect to matrix multiplication:

- (T1) It **reverses** the order of matrix multiplication: $(AB)^\top = B^\top A^\top$.
- (T2) If an $n \times n$ matrix A is invertible then so is $A^\top$, with $(A^\top)^{-1} = (A^{-1})^\top$.
- (T3) It encodes the dot product: if $\mathbf{v}, \mathbf{w} \in \mathbf{R}^n$ are viewed as $n \times 1$ (i.e., column) matrices then

$$\mathbf{v}^\top \mathbf{w} = \text{ the dot product } \mathbf{v} \cdot \mathbf{w} \text{ (considered as a } 1 \times 1 \text{ matrix).} \qquad (20.1.1)$$

Example 20.1.7. To illustrate property (T1) numerically, for the specific A and B under consideration in Example 20.1.2 let's compute $(AB)^\top$, $A^\top B^\top$, and $B^\top A^\top$. Please check for yourself that

$$(AB)^\top = \begin{bmatrix} 25 & 43 \\ 9 & 17 \end{bmatrix}, \quad A^\top B^\top = \begin{bmatrix} 16 & 9 \\ 42 & 26 \end{bmatrix}, \quad B^\top A^\top = \begin{bmatrix} 25 & 43 \\ 9 & 17 \end{bmatrix}.$$

Observe that $(AB)^\top \neq A^\top B^\top$ but $(AB)^\top = B^\top A^\top$ in this case.

A justification for (T1) in general, for those who are interested, amounts to a direct comparison of matrix entries on both sides by using the algebraic formulas defining transpose and matrix multiplication (the ij-entries are $(AB)^{\top}_{ij} = (AB)_{ji} = \sum_k a_{jk}b_{ki}$ and $(B^{\top}A^{\top})_{ij} = \sum_k (B^{\top})_{ik}(A^{\top})_{kj} = \sum_k b_{ki}a_{jk}$, and these sums are equal since the kth term in each is $a_{jk}b_{ki} = b_{ki}a_{jk}$). ■

Example 20.1.8. To illustrate property (T2), consider the matrix $A = \begin{bmatrix} 2 & -3 \\ -1 & 1 \end{bmatrix}$, so

$$A^{\top} = \begin{bmatrix} 2 & -1 \\ -3 & 1 \end{bmatrix}, \quad A^{-1} = \begin{bmatrix} -1 & -3 \\ -1 & -2 \end{bmatrix}.$$

Thus,

$$(A^{-1})^{\top} = \begin{bmatrix} -1 & -1 \\ -3 & -2 \end{bmatrix}.$$

This is indeed inverse to $A^{\top}$ (i.e., $(A^{\top})^{-1} = (A^{-1})^{\top}$ in this case), by computing a matrix product:

$$(A^{-1})^{\top}A^{\top} = \begin{bmatrix} -1 & -1 \\ -3 & -2 \end{bmatrix}\begin{bmatrix} 2 & -1 \\ -3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathrm{I}_2.$$

■

The general fact recorded in (T2) and illustrated in a special case in Example 20.1.8 that $A^{\top}$ is invertible with inverse equal to $(A^{-1})^{\top}$ for *every* invertible $n \times n$ matrix A can be seen by applying matrix transpose to the equality $A^{-1}A = \mathrm{I}_n$ and using the property (T1) of matrix transpose: in symbols,

$$A^{\top}(A^{-1})^{\top} = (A^{-1}A)^{\top} = \mathrm{I}_n^{\top} = \mathrm{I}_n.$$

If we apply the matrix transpose to the other equality $AA^{-1} = \mathrm{I}_n$ we likewise get the analogous inversion relation between the transposes in the opposite order: $(A^{-1})^{\top}A^{\top} = \mathrm{I}_n$.

Example 20.1.9. For a numerical example of (T3), consider $\mathbf{v} = \begin{bmatrix} 2 \\ -3 \\ 5 \end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix} 3 \\ 1 \\ 4 \end{bmatrix}$. Then

$$\mathbf{v}^{\top}\mathbf{w} = \begin{bmatrix} 2 & -3 & 5 \end{bmatrix}\begin{bmatrix} 3 \\ 1 \\ 4 \end{bmatrix} = [6 - 3 + 20] = [23]$$

and $\mathbf{v} \cdot \mathbf{w}$ is indeed equal to 23 (by essentially the *same* numerical calculation). ■

In general, (T3) is just a shorthand way of expressing how we multiply a $1 \times n$ matrix (one row) against an $n \times 1$ matrix (one column) to get a 1×1 matrix (a single number):

$$\begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = [v_1w_1 + v_2w_2 + \dots + v_nw_n] = [\mathbf{v} \cdot \mathbf{w}].$$

Remark 20.1.10. Expressing dot products in terms of matrix algebra as in (T3) may seem too superficial to be useful, so let us mention two applications. First, combining it with (T1) explains why the surprising Theorem 20.1.4 holds: for $\mathbf{x} \in \mathbf{R}^n$ and $\mathbf{y} \in \mathbf{R}^m$ we have

$$\mathbf{x}\cdot(A^{\top}\mathbf{y}) = (A^{\top}\mathbf{y})\cdot\mathbf{x} = (A^{\top}\mathbf{y})^{\top}\mathbf{x} = (\mathbf{y}^{\top}A^{\top\top})\mathbf{x} = (\mathbf{y}^{\top}A)\mathbf{x} = \mathbf{y}^{\top}(A\mathbf{x}) = \mathbf{y}\cdot(A\mathbf{x}) = (A\mathbf{x})\cdot\mathbf{y}. \quad (20.1.2)$$

Second, in Theorem 20.6.3 (in the optional Section 20.6) we combine (T3) with Theorem 20.1.4 to compute $\mathbf{Proj}_V$ for a linear subspace V of $\mathbf{R}^n$ using a general basis of V (not necessarily orthogonal!).

Remark 20.1.11. The quantum computing work mentioned in Example 1.4.6 involved random quantum circuits consisting of 53 "qubits"; such circuits were handled mathematically in terms of D-vectors for $D = 2^{53}$ (in accordance with rules from quantum mechanics for the circuit design). Disregarding some issues involving complex numbers, the quantum computational process was a matrix operation on D-vector circuits, expressed [**Aru2**, IV.A, (2)] as a convex combination of matrices

$$\varepsilon \mathbf{a}\mathbf{a}^\top + (1-\varepsilon)M$$

where $\mathbf{a}$ is a specific unit D-vector (a $D \times 1$ matrix, so $\mathbf{a}\mathbf{a}^\top$ is a $D \times D$ matrix) encoding "ideal output", M is a $D \times D$ matrix encoding errors, and ε is a small positive scalar. (In Example 1.4.6 we spoke of D^2-vectors rather than $D \times D$ matrices because matrix language wasn't yet available.) The expression $\mathbf{a}\mathbf{a}^\top$ for unit vectors $\mathbf{a}$ shows up *everywhere* in quantum mechanics because it computes projection to a line: $(\mathbf{a}\mathbf{a}^\top)\mathbf{x} = \mathbf{a}(\mathbf{a}^\top\mathbf{x}) = \mathbf{a}(\mathbf{a}\cdot\mathbf{x}) = (\mathbf{a}\cdot\mathbf{x})\mathbf{a} = \mathbf{Proj}_{\mathbf{a}}(\mathbf{x})$ (since $\|\mathbf{a}\| = 1$).

20.2. Properties of matrix algebra (final round). Here is a final summary:

- (From before) $A(B+C) = AB + AC$, $(A+B)C = AC + BC$, $A(BC) = (AB)C$, and $A\mathrm{I}_n = A = \mathrm{I}_m A$ for an $m \times n$ matrix A. **But** $AB \neq BA$ **in general**!
- (From before) Sometimes matrices are invertible. When they are invertible, you can multiply by their inverse to cancel them: for example, if $AB = AC$ *and A is invertible* then $B = C$, whereas if $AB = CA$ (with invertible A) then you can't conclude anything.
- (From before) Invertible matrices are always square; i.e., $n \times n$ for some n. Inversion reverses the order of multiplication: $(AB)^{-1} = B^{-1}A^{-1}$ if A and B are both invertible.
- (New) The transpose of an $m \times n$ matrix is an $n \times m$ matrix, and transpose reverses the order of multiplication: $(AB)^\top = B^\top A^\top$.
- (New) If A is invertible so is $A^\top$, and $(A^\top)^{-1} = (A^{-1})^\top$.
- (New) For $\mathbf{v}, \mathbf{w} \in \mathbf{R}^n$ viewed as $n \times 1$ matrices, the 1×1 matrix product $\mathbf{v}^\top\mathbf{w}$ equals $[\mathbf{v}\cdot\mathbf{w}]$. This yields efficient manipulation of dot products of many vectors at once via matrix algebra.

20.3. Symmetric matrices and quadratic forms. Matrix transpose is relevant to multivariable optimization via the analogue for multivariable functions $f(x_1, \dots, x_n)$ of the second-derivative test from single-variable calculus. As a warm-up to this, let's first recall that second-derivative test.

In single-variable calculus, at a critical point a of a function $f : \mathbf{R} \to \mathbf{R}$ (i.e., $f'(a) = 0$), the second derivative $f''(a)$ controls the behavior of f near a when $f''(a) \neq 0$. The point is that the degree-2 Taylor approximation to f at a says that for x near a we have

$$f(x) \approx f(a) + f'(a)(x-a) + f''(a)(x-a)^2/2! = f(a) + (1/2)f''(a)(x-a)^2$$

(since $f'(a) = 0$), and so the sign of $f''(a)$ determines whether the value of $f(x)$ for x near (but different from) a is $> f(a)$ (when $f''(a) > 0$) or is $< f(a)$ (when $f''(a) < 0$). This is the *second-derivative test*. We have used our knowledge of the behavior of the function ct^2 for nonzero c: a concave-up parabola for $c > 0$ (with a *minimum* at $t = 0$), and a concave-down parabola for $c < 0$ (with a *maximum* at $t = 0$).

What is the analogue for a function $f(x_1, \dots, x_n)$ of n variables near a point $\mathbf{a} = (a_1, \dots, a_n)$ for which all partial derivatives $(\partial f/\partial x_i)(\mathbf{a})$ vanish? In Chapters 25–26 we will take up this topic (a "multivariable second-derivative test"). The role of parabolas ct^2 in the single-variable case will be replaced by functions such as $7t_1^2 - 2t_2^2 + 3t_1t_2$ for $n = 2$ and $8t_1^2 + 5t_2^2 - t_3^2 + 2t_1t_2 + 7t_1t_3 - t_2t_3$ for $n = 3$.

Definition 20.3.1. A function of the type $q(t_1, \dots, t_n) = c_1t_1^2 + \dots + c_nt_n^2 + \sum_{i<j} c_{ij}t_it_j$ is called a *quadratic form* (since it involves products of two variables at a time).

Example 20.3.2. The variance of a (linear) investment portfolio is a quadratic form in the weights of the assets in the portfolio [AI, (I.2.23)], and various types of energy in *multi-part* physical systems are quadratic forms in positions or velocities (see Example 20.3.14). ■

Example 20.3.3. The nicest quadratic form of all is $t_1^2 + \cdots + t_n^2$ that computes the squared-length of vectors in $\mathbf{R}^n$. If we apply a linear change of coordinates then the formula for squared-length in the new coordinates is still a quadratic form but typically different from "the sum of squares of coordinates".

For example, consider describing vectors in $\mathbf{R}^2$ using the coordinates (x', y') arising from the basis $\mathbf{v}_1' = 2\mathbf{e}_1 + \mathbf{e}_2$ and $\mathbf{v}_2' = \mathbf{e}_1 + \mathbf{e}_2$ (corresponding to a linear transformation of the plane distorting distances and angles). The usual squared-length is expressed in terms of (x', y') as

$$\begin{aligned}\|x'\mathbf{v}_1' + y'\mathbf{v}_2'\|^2 = \|x'(2\mathbf{e}_1 + \mathbf{e}_2) + y'(\mathbf{e}_1 + \mathbf{e}_2)\|^2 &= \|(2x' + y')\mathbf{e}_1 + (x' + y')\mathbf{e}_2\|^2 \\ &= (2x' + y')^2 + (x' + y')^2 \\ &= 5x'^2 + 6x'y' + 2y'^2.\end{aligned}$$

This is a quadratic form but looks very different from $x'^2 + y'^2$! ■

The geometry of level sets $Q(\mathbf{x}) = c$ of quadratic forms Q in n variables is quite rich beyond the case $n = 1$ (for $n = 2$ it is the story of ellipses and hyperbolas). It turns out that the data of a quadratic form in n variables can be encoded in terms of a special class of $n \times n$ matrices called "symmetric". We shall now discuss this link (using transposes extensively), the significance of which will be explained in Chapter 26.

Example 20.3.4. Consider the matrices

$$M = \begin{bmatrix} 2 & -3 \\ -3 & 1 \end{bmatrix}, \quad N = \begin{bmatrix} 1 & 0 & 7 \\ 0 & -2 & 3 \\ 7 & 3 & -1 \end{bmatrix}, \quad N' = \begin{bmatrix} 1 & 0 & 7 \\ 0 & -2 & 3 \\ 7 & 2 & -1 \end{bmatrix}.$$

Check that $M^\top = M$, $N^\top = N$, and $N'^\top \neq N'$. If you think about the process of computing these transposes, the features of M and N that make them equal to their own transpose but not so for N' is a symmetry with respect to flipping across the diagonal: for each of M and N the $(1,2)$-entry is equal to the $(2,1)$-entry, and in general the ij-entry is equal to the ji-entry for all i, j, whereas for N' the (2,3)-entry and the (3,2)-entry are not the same. ■

The property of M and N above has a special name, already mentioned in Theorem 20.1.4:

Definition 20.3.5. A matrix A is called *symmetric* if $A^\top = A$ (or in other words, the ij-entry and the ji-entry coincide for all i, j). Informally, A is "symmetric around its diagonal".

A symmetric matrix is always square. (For example, if A is a 3×5 matrix then $A^\top$ is a 5×3 matrix and so certainly A *cannot* be equal to $A^\top$.) As further illustrations, $\begin{bmatrix} 1 & 3 \\ 3 & 4 \end{bmatrix}$ is symmetric but $\begin{bmatrix} 1 & 3 \\ 5 & 4 \end{bmatrix}$ isn't symmetric. Also, for any $f : \mathbf{R}^n \to \mathbf{R}$ and $\mathbf{a} \in \mathbf{R}^n$, in Example 13.5.6 we saw that the $n \times n$ derivative matrix $(D(\nabla f))(\mathbf{a})$ is symmetric due to equality of mixed partials.

Example 20.3.6. Suppose M is a symmetric $n \times n$ matrix (so $M^\top = M$). Then $(M^2)^\top = M^\top M^\top = MM = M^2$, so M^2 is symmetric. For a 3×3 example,

$$A = \begin{bmatrix} 2 & 13 & -3 \\ 13 & 5 & 7 \\ -3 & 7 & 11 \end{bmatrix} \text{ implies } A^2 = \begin{bmatrix} 182 & 70 & 52 \\ 70 & 243 & 73 \\ 52 & 73 & 179 \end{bmatrix},$$

so A^2 is also seen to be symmetric by inspection. But this goes further: *every* power M^r is symmetric. For example, since $M^3 = MM^2$ with each of M and M^2 symmetric, we have $(M^3)^\top = (MM^2)^\top = (M^2)^\top M^\top = M^2M = M^3$, so M^3 is symmetric. Likewise, $M^4 = MM^3$ with both M and M^3 symmetric, so $(M^4)^\top = (MM^3)^\top = (M^3)^\top M^\top = M^3M = M^4$, so M^4 is symmetric.

The same kind of argument can be carried out repeatedly to apply to *all higher powers* of M. For the specific 3×3 matrix A above, it is much easier to check symmetry of A^3 and A^4 by the preceding short conceptual argument for all M than to compute them explicitly, but in case you're curious:

$$A^3 = \begin{bmatrix} 1118 & 3080 & 516 \\ 3080 & 2636 & 2294 \\ 516 & 2294 & 2324 \end{bmatrix}, \quad A^4 = \begin{bmatrix} 40728 & 33546 & 23882 \\ 33546 & 69278 & 34446 \\ 23882 & 34446 & 40074 \end{bmatrix}.$$

■

Example 20.3.7. If M is an invertible $n \times n$ matrix then $M^\top$ is always invertible with inverse $(M^\top)^{-1}$ equal to $(M^{-1})^\top$ (as we saw in Section 20.1). In the special case that M is also symmetric, it follows that the $n \times n$ matrix M^{-1} is also symmetric. In words:

> The inverse of a symmetric invertible matrix is always symmetric.

Indeed, $(M^{-1})^\top = (M^\top)^{-1} = M^{-1}$, verifying the very definition of symmetry for M^{-1}. To see this in an explicit example, one can check for the symmetric 3×3 matrix A in Example 20.3.6 that the matrix

$$B = \begin{bmatrix} -3/1219 & 82/1219 & -1/23 \\ 82/1219 & -13/2438 & 1/46 \\ -1/23 & 1/46 & 3/46 \end{bmatrix}.$$

which is symmetric by inspection is inverse to A (e.g., compute that AB or BA is equal to I_3). ■

In Example 20.1.6 we noticed for some specific 2×2 matrices M that the matrix products $M^\top M$ and $M\,M^\top$ are actually symmetric. This is a general fact:

> **Theorem 20.3.8.** For *any* $m \times n$ matrix M, the $n \times n$ matrix $M^\top M$ and the $m \times m$ matrix $M\,M^\top$ (called *Gram* matrices) are symmetric.

Remark 20.3.9. One way to understand the symmetry of $M^\top M$ is to think about what its ij-entry means: it is the dot product of the ith row of $M^\top$ and the jth column of M, but the rows of $M^\top$ are the columns of M by another name! Thus the ij-entry of $M^\top M$ is the dot product $\mathbf{m}_i \cdot \mathbf{m}_j$ of the ith and jth columns of M, so symmetry of $M^\top M$ (i.e., insensitivity of its ij-entry to replacing ij with ji) expresses the commutativity of dot products. The same holds for $M\,M^\top$ by using rows instead of columns. (A slicker explanation of symmetry is the calculation $(M^\top M)^\top = M^\top (M^\top)^\top = M^\top M$, and similarly for $M\,M^\top$.)

Yet another class of examples arises from the following result (which is proved in Remark B.3.2):

Proposition 20.3.10. For a linear subspace V of $\mathbf{R}^n$, the $n \times n$ matrix for $\mathbf{Proj}_V : \mathbf{R}^n \to \mathbf{R}^n$ is symmetric.

Our interest in symmetric matrices is partly due to their role in the following important construction. Let A be a *symmetric* $n \times n$ square matrix (such as $A = M^\top M$ for any $m \times n$ matrix M). For any vector $\mathbf{v} \in \mathbf{R}^n$, the product $\mathbf{v}^\top A\mathbf{v}$ is the 1×1 matrix with entry $\mathbf{v} \cdot (A\mathbf{v})$. This construction will arise a lot in the rest of the book, so we give it a special notation: define the function $q_A : \mathbf{R}^n \to \mathbf{R}$ by the rule

$$q_A(\mathbf{v}) = \mathbf{v} \cdot (A\mathbf{v}) = \mathbf{v}^\top A\mathbf{v} \quad \text{for } \mathbf{v} \in \mathbf{R}^n. \tag{20.3.1}$$

Let's see what this function looks like in some examples, and then see the relevance of symmetry for A.

Example 20.3.11. For the symmetric matrices

$$A = \begin{bmatrix} 5 & -3 \\ -3 & 7 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 13 & -3 \\ 13 & 5 & 7 \\ -3 & 7 & 11 \end{bmatrix}$$

check that

$$q_A\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x \\ y \end{bmatrix}^\top A \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x & y \end{bmatrix} A \begin{bmatrix} x \\ y \end{bmatrix} = 5x^2 + 7y^2 - 6xy,$$

$$q_B\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} x \\ y \\ z \end{bmatrix}^\top B \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x & y & z \end{bmatrix} B \begin{bmatrix} x \\ y \\ z \end{bmatrix} = 2x^2 + 5y^2 + 11z^2 + 26xy - 6xz + 14yz.$$ ■

More generally, if A is a symmetric $n \times n$ matrix, the function $q_A(\mathbf{x})$ for $\mathbf{x} \in \mathbf{R}^n$ always looks like a sum of terms involving either x_i^2 or $x_i x_j$, which is to say q_A is a *quadratic form* (as in Definition 20.3.1).

It is worth memorizing how this works out for $n = 2, 3$, rather than multiplying explicitly each time:

$$\begin{bmatrix} x \\ y \end{bmatrix}^\top \begin{bmatrix} a & u \\ u & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = ax^2 + by^2 + 2uxy, \tag{20.3.2}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}^\top \begin{bmatrix} a & u & v \\ u & b & w \\ v & w & c \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = ax^2 + by^2 + cz^2 + 2uxy + 2vxz + 2wyz \tag{20.3.3}$$

(as you can check by multiplying these out just once). This recovers Example 20.3.11 as special cases.

The general pattern in $q_A(\mathbf{x})$ for $n \times n$ symmetric $A = (c_{ij})$ is that the diagonal entry c_{ii} multiplies against x_i^2 and the off-diagonal entry $c_{ij} = c_{ji}$ multiplies against $x_i x_j$ up to an additional factor of 2. For instance, with $n = 3$ as shown in (20.3.3), the diagonal entries multiply against the corresponding squares x^2, y^2, z^2 while the common $(1, 2)$-entry and $(2, 1)$-entry is multiplied against xy up to a factor of 2, the common $(1, 3)$-entry and $(3, 1)$-entry is multiplied against xz up to a factor of 2, and the common $(2, 3)$-entry and $(3, 2)$-entry is multiplied against yz up to a factor of 2.

For any symmetric $n \times n$ matrix, if we know the diagonal entries c_{ii} and the entries c_{ij} above the diagonal (i.e., $i < j$) then we know the entries below the diagonal due to the symmetry $c_{ji} = c_{ij}$; look at the examples A and B in Example 20.3.11 to see this explicitly. But a quadratic form $Q(x_1, \ldots, x_n)$ in n variables is determined by the "same" amount of data: the coefficients of each x_i^2 and of each $x_i x_j$ for $i < j$. This allows us to find a **unique** symmetric A for which $Q = q_A$.

Example 20.3.12. In Example 20.3.11, from q_B we can read off B: the coefficients of x^2, y^2, z^2 tell us the diagonal entries of B, and the coefficients of the cross-terms xy, xz, yz tell us the off-diagonal entries of B up to dividing by 2. **We can carry out this reverse process for any quadratic form** $Q(x_1, \ldots, x_n)$**.**

As an illustration, for $Q(x, y, z) = 8x^2 + 5y^2 - z^2 + 2xy + 7xz - yz$ we seek a symmetric 3×3 matrix M for which $q_M = Q$. There is exactly one such M, namely:

$$M = \begin{bmatrix} 8 & 1 & 7/2 \\ 1 & 5 & -1/2 \\ 7/2 & -1/2 & -1 \end{bmatrix}.$$

Using (20.3.3), check for yourself that indeed $q_M = Q$.

In general, for any quadratic form $Q(\mathbf{x})$ in n variables $x_1, \ldots, x_n$, we have $Q = q_A$ for the symmetric $n \times n$ matrix A whose ii-entry is the coefficient of x_i^2 in Q and whose ij-entry and ji-entry are each *half* the coefficient of $x_i x_j$ in Q. A financial context with this reverse process for big n is in [Al, I.2.4.1-I.2.4.2], where: $\mathbf{x}$ is a vector of portfolio weights, Q is the variance of the returns, and A is the covariance matrix of returns. A physical context for $n = 3$ is in [**Feyn1**, II, (31.1)-(31.8)], where: $\mathbf{x}$ is an electric field, Q is the energy density of a polarized crystal, and A is the "polarization tensor" of the crystal.

For an example with $n = 2$, if $Q(x, y) = 3x^2 + 2y^2 - 5xy$ then what is the symmetric 2×2 matrix B for which $q_B = Q$? Check via (20.3.2) that $B = \begin{bmatrix} 3 & -5/2 \\ -5/2 & 2 \end{bmatrix}$. ■

Remark 20.3.13. In 1968, Lars Onsager got the Nobel Prize in Chemistry for establishing the symmetry of a matrix L of "kinetic coefficients" for many irreversible thermodynamic processes. The generality of this *Onsager reciprocal relation* (or the "4th Law of Thermodynamics") was unexpected. The value $q_L(\mathbf{x})$ is the rate of entropy change in near-equilibrium situations for $\mathbf{x}$ encoding small thermodynamic forces, so $q_L(\mathbf{x}) \geq 0$ for small $\mathbf{x}$ (by the 2nd Law of Thermodynamics) and hence all $\mathbf{x}$ (since $q_L(\mathbf{x}) = N^2 q_L(\mathbf{x}/N)$).

The passage between symmetric matrices and quadratic forms is essentially linguistic so far. Example 20.3.14 below illustrates how quadratic forms are relevant to some mechanical situations, due to energies such as "$(1/2)mv^2$" (kinetic energy) and "$(1/2)kx^2$" (elastic energy) in physics being replaced by multivariable quadratic forms for energies of mechanical systems consisting of many parts.

Example 20.3.14. The passage between symmetric matrices and quadratic forms arises in mechanical engineering for vibrational mechanics (e.g., a moving car or other system involving many spring-like components), structural analysis for strength of materials (with beams, etc.), and so on. For such a system consisting of many parts interacting in a *linear* manner (as is often the case, though some phenomena such as cracking tend to be non-linear), the differential equations describing the positions of the parts as functions of time are most efficiently expressed in vector form using several $n \times n$ matrices M, K, C respectively called the *mass matrix*, the *stiffness matrix*, and the *damping matrix*.

Here, n is the number of "degrees of freedom" for vibrational motion in the system (i.e., the number of "independent" types of motion). It is determined by inspection of the physical structure. For a system of masses linked among each other via springs that all move along the same line, this n is the number of masses. The matrices M, K, and C are symmetric: for M this is by its definition, but for K it is due to a physical reason called the Maxwell-Betti reciprocal work theorem.

Up to a factor of 1/2, the quadratic forms associated with these symmetric matrices encode specific types of energy in the mechanical system, as we list below. In particular, one could regard each type of energy as a means of *encoding* M, K, and C since a symmetric matrix can always be recovered from the corresponding quadratic form via the pattern as in (20.3.2) and (20.3.3). This goes beyond mere mathematical language: *the ability to express energies in physical settings in terms of specific symmetric matrices (via quadratic forms) encodes deeper properties of the mechanics once know more about the remarkable mathematics of symmetric matrices*, as we shall see in Example 24.6.4.

To express the link between quadratic forms and energy, we need to use a vector $\mathbf{x}(t)$ recording the positions of all parts at time t as *displacements* from the rest position (so $\mathbf{x}(0) = \mathbf{0}$); this is an n-vector where n is the number of "degrees of freedom" for vibrational motion (as mentioned above). The componentwise derivative $\mathbf{x}'(t)$ encodes velocities of all parts at time t.

- The mass matrix M encodes masses of all parts, and $(1/2)q_M(\mathbf{x}'(t)) = (1/2)\mathbf{x}'(t)^\top M\mathbf{x}'(t)$ is the total kinetic energy (generalizing $(1/2)mv^2$ from physics).
- The stiffness matrix K is a higher-dimensional version of the spring constant k from Hooke's Law in physics. It encodes the elastic potential energy in the system, and $(1/2)q_K(\mathbf{x}(t)) = (1/2)\mathbf{x}(t)^\top K\mathbf{x}(t)$ is the total elastic potential energy of the system (generalizing $(1/2)kx^2$ from physics with springs).

- For the damping matrix C, $(1/2)q_C(\mathbf{x}'(t)) = (1/2)\mathbf{x}'(t)^\top C\mathbf{x}'(t)$ is the total dissipated energy (energy lost to do more work in the mechanical system; e.g., via heat loss or other means). ■

20.4. Orthogonal matrices. The significance of the link between symmetric matrices and quadratic forms for a multivariable second-derivative test (to determine when a critical point is a local extremum) is discussed in Chapter 26. For now, we use quadratic forms to introduce a special class of matrices whose geometric meaning is visualized in terms of "rigid motions" of space (as in a flight simulator) but has practical importance far beyond rigid motions (for solving systems of linear equations, principal component analysis of data, discrete Fourier transform of signals, etc.). We observed in Example 14.5.2 that the columns of a rotation matrix are orthogonal to one another and have length 1. In fact, the rows are also orthogonal to one another and have length 1, and we can easily invert any rotation matrix! To explain this, we'll use the matrix transpose and the link between symmetric matrices and quadratic forms.

Suppose R is the matrix of a rotation of $\mathbf{R}^3$ around the origin. Note that $R\mathbf{v}$ must have the same length as $\mathbf{v}$ for any $\mathbf{v} \in \mathbf{R}^3$, since one is a rotation of the other. That is: $\|R\mathbf{v}\| = \|\mathbf{v}\|$. Squaring gives $\|R\mathbf{v}\|^2 = \|\mathbf{v}\|^2$; expressing this in terms of dot products written as matrix products says $\mathbf{v}^\top\mathbf{v} = (R\mathbf{v})^\top(R\mathbf{v})$. Using the rules for transpose and matrix multiplication on the right side yields

$$\mathbf{v}^\top\mathbf{v} = (\mathbf{v}^\top R^\top)(R\mathbf{v}) = \mathbf{v}^\top(R^\top R)\mathbf{v} = q_{R^\top R}(\mathbf{v}) \tag{20.4.1}$$

for *all* $\mathbf{v} \in \mathbf{R}^3$, where the quadratic form on the right is associated with the matrix product $R^\top R$.

For a 3×3 matrix R, one way to guarantee that the equality $\mathbf{v}^\top\mathbf{v} = q_{R^\top R}(\mathbf{v})$ holds for all $\mathbf{v}$ is that $R^\top R = \mathrm{I}_3$ (the 3×3 identity matrix), since $q_{\mathrm{I}_3}(\mathbf{v}) = x^2 + y^2 + z^2$. But in fact this is the *only* way that (20.4.1) can hold for all $\mathbf{v} \in \mathbf{R}^3$! Indeed, we saw rather concretely in Example 20.3.12 that a quadratic form in n variables arises from a *unique* symmetric $n \times n$ matrix, so since the quadratic form $Q(\mathbf{x}) = \|\mathbf{x}\|^2$ is equal to $q_{\mathrm{I}_n}(\mathbf{x})$ for all $\mathbf{x} \in \mathbf{R}^n$ it follows that the *only* symmetric $n \times n$ matrix A for which $q_A(\mathbf{x}) = \|\mathbf{x}\|^2$ for all $\mathbf{x} \in \mathbf{R}^n$ is $A = \mathrm{I}_n$. Since $R^\top R$ is always symmetric (for *any* 3×3 matrix R), the condition $q_{R^\top R}(\mathbf{v}) = \|\mathbf{v}\|^2$ for all $\mathbf{v} \in \mathbf{R}^3$ *forces* $R^\top R = \mathrm{I}_3$, as claimed.

But the condition $R^\top R = \mathrm{I}_n$ for an $n \times n$ matrix R says exactly that $R^\top$ and R are inverse matrices (due to the non-obvious Theorem 18.1.8), so we also get $RR^\top = \mathrm{I}_n$. This reasoning is very special – as we saw in Example 20.1.6, in general we *do not* have $M^\top M = MM^\top$ for general square matrices M.

Any rotation matrix R on $\mathbf{R}^3$ satisfies $R^\top R = \mathrm{I}_3$. Equivalently: the inverse of R is its transpose $R^\top$.

Let us spell out the condition $R^\top R = \mathrm{I}_n$ for a general $m \times n$ matrix R. The computation (20.4.1) still works and shows that this is *exactly* the condition that $\|R\mathbf{v}\|^2 = \|\mathbf{v}\|^2$ for all $\mathbf{v} \in \mathbf{R}^n$ (equivalently, $T_R : \mathbf{R}^n \to \mathbf{R}^m$ is a length-preserving linear transformation). But the ij-entry of $R^\top R$ is the dot product of the ith and jth columns of R (see Remark 20.3.9), so the fact that $R^\top R = \mathrm{I}_n$ translates to the condition that *the columns of R are orthonormal* (i.e., each column has length 1 and the dot product of any two distinct columns vanishes). Thus, for $m = n$ the non-obvious consequence noted above that then $RR^\top = \mathrm{I}_n$ translates to *the rows of R being orthonormal.*

The n columns of R are linearly independent m-vectors due to being an orthonormal collection, so their span in $\mathbf{R}^m$ has dimension n and hence $n \leq \dim \mathbf{R}^m = m$; i.e, R is a "tall" matrix. To summarize:

Theorem 20.4.1. The following conditions are equivalent, for a general $m \times n$ matrix A.

- The associated linear transformation $T_A : \mathbf{R}^n \to \mathbf{R}^m$ is length-preserving.
- $A^\top A = \mathrm{I}_n$.
- (only when $m = n$) $AA^\top = \mathrm{I}_n$.

- The n columns of A are an orthonormal collection of m-vectors (so $n \leq m$; i.e., A is "tall").
- (only when $m = n$) The rows of A are an orthonormal collection.

If these hold then T_A is also angle-preserving, so when $m = n$ it is a "rigid motion" of $\mathbf{R}^n$ fixing $\mathbf{0}$.

The reason for angle-preservation when the other equivalent conditions hold is that angles are determined by dot products and lengths (see Definition 2.1.6) yet dot products in turn can be computed in terms of suitable lengths and vector sums:

$$\mathbf{x} \cdot \mathbf{y} = \frac{1}{2}((\mathbf{x}+\mathbf{y}) \cdot (\mathbf{x}+\mathbf{y}) - \mathbf{x} \cdot \mathbf{x} - \mathbf{y} \cdot \mathbf{y}) = \frac{1}{2}(\|\mathbf{x}+\mathbf{y}\|^2 - \|\mathbf{x}\|^2 - \|\mathbf{y}\|^2). \tag{20.4.2}$$

Hence, since T_A respects sums, when it preserves lengths it must preserve dot products and so also angles!

Definition 20.4.2. A square matrix A satisfying the equivalent conditions in Theorem 20.4.1 is called an *orthogonal* matrix. (For example, the matrix of any rotation in $\mathbf{R}^3$ fixing the origin is an orthogonal matrix. "Fixing the origin" cannot be dropped: without this, linearity fails.)

You might think such a matrix should be called "orthonormal" rather than "orthogonal", but the terminology in Definition 20.4.2 has been used in math and its applications throughout natural sciences, computer science, and data analysis for a very long time and so we are stuck with it: *an orthogonal matrix is a square matrix with orthonormal columns*. (There is no special name for a square matrix with merely pairwise orthogonal columns, and this latter concept turns out to be useless.)

Example 20.4.3. In Section 14.5 the matrix

$$C = \begin{bmatrix} 1/2 & -1/\sqrt{2} & 1/2 \\ 1/2 & 1/\sqrt{2} & 1/2 \\ -1/\sqrt{2} & 0 & 1/\sqrt{2} \end{bmatrix}$$

was made by composing two rotations (see (14.5.2)), and in Example 14.5.2 it was discussed in two ways (computational and geometric) that its columns are orthonormal. Theorem 20.4.1 says that its rows are orthonormal: check this by direct computation. Orthonormality of the rows doesn't have the same vivid geometric meaning as does orthonormality of the columns. ■

Any "rigid motion" of $\mathbf{R}^n$ preserving the origin *must* be a linear transformation, by the characterization of such operations in Section 14.2. Hence, by Theorem 20.4.1 such operations are the effect of an orthogonal matrix. Orthogonal $n \times n$ matrices therefore express "rigid motions" of $\mathbf{R}^n$ fixing the origin.

Proposition 20.4.4. It is *really easy* to invert an orthogonal matrix: $A^{-1} = A^\top$. Moreover, if A and B are orthogonal $n \times n$ matrices then both AB and BA are orthogonal.

The first assertion in Proposition 20.4.4 expresses one of the several equivalent characterizations of orthogonality in the square case (namely: $A^\top A = \mathrm{I}_n$). To illustrate the second assertion, if A and B are matrices of rotations on $\mathbf{R}^3$ fixing $\mathbf{0}$ then BA represents the rigid motion obtained by doing first A and then B, and AB is the other way around. In general, it holds because if T_A and T_B are length-preserving then so are $T_A \circ T_B = T_{AB}$ and $T_B \circ T_A = T_{BA}$, so AB and BA are orthogonal.

Orthogonal matrices arise in numerical analysis, including efficient matrix algebra with $m \times n$ matrices for large m and n via the "QR-decomposition" to be discussed in Chapter 22, and underlie the singular value decomposition of matrices (which will be introduced in Section 27.3 and has tons of applications throughout computer science, (e.g., data compression), signal processing, and much more).

Example 20.4.5. Returning to the orthogonal 3×3 matrix C in Example 20.4.3, now we can write down the inverse of C, since it is just $C^\top$. Check by direct matrix multiplication that $C^\top$ really is inverse to C, and observe that computing this matrix product amounts to exactly computing dot products among rows (or among columns) of C. Hence, the equality $C^{\top}C = \mathrm{I}_3$ encodes precisely the orthogonality of C. ■

20.5. Application I: discrete Fourier transform. One of the most important applications of orthogonal $n \times n$ matrices is the *discrete Fourier transform* (DFT), which is widely used in science and engineering. The typical context is working with signals (or other continuous data) measured at regular time intervals. The resulting collection of n measurements are arranged as entries in an n-vector. What is of most interest is how each entry in the n-vector varies from the adjacent ones. The idea behind DFT is to identify a useful basis of n-vectors corresponding to oscillations at specific frequencies (the extreme cases being no oscillation or alternating signs: the n-vectors whose entries are all 1 or alternating ± 1).

When the n-vector of data is expressed as a superposition (i.e., linear combination) of the magical DFT basis, the coefficients that arise (in effect, the amplitude with which each of the "basis frequencies" occurs in the data vector) can provide extremely useful information about the signal that is difficult to perceive by direct inspection of the data n-vector. Some among the endless list of applications include:

- spectroscopy to extract information about matter from measurements of electromagnetic radiation (e.g., using light waves to determine the structure of molecules and the composition of stars, MRI to determine spatial layout of human organs without surgery by harnessing density measurements along lines, EEG's for measuring heart activity and the design of pacemakers, and gravity waves to analyze the merging of black holes),
- conversion of visual or audio information into electronic form, including video on portable electronic devices, smartphone communication, and MP3 files for digital music,
- FM (frequency modulation) and AM (amplitude modulation) radio involve the manipulation ("modulation") of frequencies or amplitudes of wave components of audio signals to ensure that different station's signals can transmit without interfering with each other,
- denoising an audio signal or smoothing an image.

In many practical applications n is in the hundreds, thousands, or beyond. This poses a significant problem for the utility of the discrete Fourier transform F_n on n-vectors because from its definition (given below, as essentially n sums of n products), the time required to compute $F_n(\mathbf{x})$ is on the order of n^2. But in 1965 something amazing happened: the discovery of the fast Fourer Transform (FFT), an algorithm that computes F_n in time around $n \log_2 n$ (so basically around the size n of the data, since $\log_2 n$ grows very slowly in n). That dramatic speed-up by a factor of around n times as fast has been essential in everything we take for granted in electronics, and is the reason that FFT is on the list of "Top Ten Algorithms of the (20th) Century" [DS]. FFT involves a lot of partitioning of the data into halves, so in practice usually n is a power of 2 (and in particular, it is even).

In physical terms, it turns out that the power or energy of a signal $\mathbf{x}$ (which is $\mathbf{x} \cdot \mathbf{x}$ up to a scaling factor) is identical to that of its DFT. By the equivalence of the first and second properties in Theorem 20.4.1, the matrix computing DFT is therefore *orthogonal*. This has tremendous practical importance because it allows one to *invert* the transform very easily since inversion for orthogonal matrices M is so simple: $M^{-1} = M^{\top}$ (Proposition 20.4.4). For instance, removing high-frequency terms beyond human hearing range in the transform of an audio signal is a compression of the data for which inversion of F_n produces the *same sound* to human ears.

The standard definition of DFT involves complex numbers and so to convey the idea using real numbers, we'll give a variant encoding the same information. (There are other real-valued variants, such as the discrete cosine transform that is widely used for data compression.) In terms of linear transformations, for even $n > 2$ (practical examples have n in the hundreds or more) the *discrete Fourier transform* (DFT) $F_n : \mathbf{R}^n \to \mathbf{R}^n$ carries $(x_0, \dots, x_{n-1}) \in \mathbf{R}^n$ to $(y_0, \dots, y_{n-1})$ defined by:

$$y_0 = \frac{1}{\sqrt{n}}(x_0 + \cdots + x_{n-1}), \qquad y_j = \frac{1}{\sqrt{n}}\sum_{k=0}^{n-1}\sqrt{2}\cos((2\pi jk)/n)x_k \quad \text{for } 1 \le j < n/2,$$

$$y_{n/2} = \frac{1}{\sqrt{n}}\sum_{k=0}^{n-1}(-1)^k x_k, \qquad y_j = \frac{1}{\sqrt{n}}\sum_{k=0}^{n-1}\sqrt{2}\sin((2\pi jk/n)x_k \quad \text{for } n/2 < j \le n-1.$$

The definition of F_n looks complicated only because it is a real-number reformulation of something simpler in terms of complex numbers. The fact that F_n is orthogonal is mysterious with the above formulation but is more transparent in the complex-number version. The ugly factor $1/\sqrt{n}$ ensures that $F_n^\top F_n$ is the $n \times n$ identity matrix I_n (it would be $n\mathrm{I}_n$ otherwise).

As an illustration for the special case $n = 8$, the matrix of F_8 is

$$\frac{1}{\sqrt{8}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \sqrt{2} & 1 & 0 & -1 & -\sqrt{2} & -1 & 0 & 1 \\ \sqrt{2} & 0 & -\sqrt{2} & 0 & \sqrt{2} & 0 & -\sqrt{2} & 0 \\ \sqrt{2} & -1 & 0 & 1 & -\sqrt{2} & 1 & 0 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 0 & -1 & \sqrt{2} & -1 & 0 & 1 & -\sqrt{2} & 1 \\ 0 & -\sqrt{2} & 0 & \sqrt{2} & 0 & -\sqrt{2} & 0 & \sqrt{2} \\ 0 & -1 & -\sqrt{2} & -1 & 0 & 1 & \sqrt{2} & 1 \end{bmatrix}$$

20.6. Application II: higher-degree curve fitting and projections via matrix algebra (optional). As an application of the efficiency of the language of matrix algebra (such as matrix inverse and transpose), we consider the problem of "least-squares" fitting of a curve to data points. Rather than finding a straight line $y = mx + b$ that best fits n given data points (x_i, y_i), sometimes one may seek a higher-degree "best-fit" curve such as a parabola $y = ax^2 + bx + c$ (as occurs for some models of population growth as well as in physics). Similarly to Sections 7.3 and 10.3 that addressed a best-fit line, for a best-fit parabola we seek to minimize the "sum of squared error" function

$$E(a, b, c) = \sum_i (y_i - (ax_i^2 + bx_i + c))^2.$$

Let's express this minimization problem in the geometric language of vectors. Similarly to the notation in Section 7.3, form the associated vectors

$$\mathbf{1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{X}' = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \vdots \\ x_n^2 \end{bmatrix}$$

in $\mathbf{R}^n$. The desired triple (a, b, c) minimizes $E(a, b, c) = \|\mathbf{Y} - (a\mathbf{X}' + b\mathbf{X} + c\mathbf{1})\|^2$, which is the same as minimizing the distance $\|\mathbf{Y} - (a\mathbf{X}' + b\mathbf{X} + c\mathbf{1})\|$. In other words, we seek the point $a\mathbf{X}' + b\mathbf{X} + c\mathbf{1}$ in $\mathrm{span}(\mathbf{X}', \mathbf{X}, \mathbf{1})$ that is as close as possible to $\mathbf{Y}$.

Assume we are not in the degenerate situation where the data points all lie on one or two vertical lines (i.e., there are at least 3 different values among the x_i's). Then it can be shown that the vectors $\mathbf{X}'$, $\mathbf{X}$, and $\mathbf{1}$ are linearly independent (if you are curious about the reason for this, which rests on determinants, see Section E.5 beginning at Theorem E.5.5), so our task is an instance of the following slightly more general problem. For $\mathbf{Y} \in \mathbf{R}^n$ suppose we're given linearly independent $\mathbf{v}_1, \dots, \mathbf{v}_m \in \mathbf{R}^n$ and want to *find the point* $\mathbf{Y}^*$ *in* $\mathrm{span}(\mathbf{v}_1, \dots, \mathbf{v}_m)$ *closest to the point* $\mathbf{Y}$. (The assumption that the $\mathbf{v}_i$'s are linearly independent is very reasonable: it just says that the span is described without redundancy.) The motivating situation is the special case $m = 3$ with $\mathbf{v}_1 = \mathbf{1}$, $\mathbf{v}_2 = \mathbf{X}$, $\mathbf{v}_3 = \mathbf{X}'$.

In Section 6.2 we discussed the *geometric* problem of finding the point in a linear subspace W of $\mathbf{R}^n$ closest to a given point $\mathbf{x} \in \mathbf{R}^n$, where we called it the *projection* of $\mathbf{x}$ onto W, so

$$\mathbf{Y}^* = \mathbf{Proj}_{\mathrm{span}(\mathbf{v}_1,\dots,\mathbf{v}_m)} \mathbf{Y}.$$

Computing that projection can be done using an orthogonal basis for the subspace, which we learned how to build in Chapter 19. Our next aim is to give a *new* way of computing it via using matrix algebra involving the transpose; the resulting formula for the projection will *not* require having an orthogonal basis (so the given basis $\mathbf{v}_1, \dots, \mathbf{v}_m$ may be used "as is", whatever it may be).

For the discussion that follows, we will illustrate everything by carrying along an example with $m = 2$ and $n = 3$:

$$\mathbf{v}_1 = \begin{bmatrix} 3 \\ -2 \\ 1 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \\ 4 \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} -2 \\ 1 \\ 4 \end{bmatrix}. \tag{20.6.1}$$

We seek the point $\mathbf{Y}^* \in \mathbf{R}^3$ in the plane $\mathrm{span}(\mathbf{v}_1, \mathbf{v}_2)$ closest to $\mathbf{Y}$. Although it can be computed as the projection of $\mathbf{Y}$ into this plane, here we give a different approach to computing $\mathbf{Y}^*$ via matrix algebra.

We shall first give the general formula (which *you should not memorize*) and then work out what it says in the above numerical setting. Let's arrange the vectors $\mathbf{v}_1, \dots, \mathbf{v}_m$ (that we assumed to be linearly independent) as the columns of a single $n \times m$ matrix $\mathbf{V}$ defined as follows:

$$\mathbf{V} = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m \\ | & | & \cdots & | \end{bmatrix} \tag{20.6.2}$$

For the running numerical example in (20.6.1), this is the 3×2 matrix $\begin{bmatrix} 3 & 0 \\ -2 & 1 \\ 1 & 4 \end{bmatrix}$.

The transpose matrix $\mathbf{V}^\top$ has its rows given by the columns of $\mathbf{V}$ that are just the $\mathbf{v}_i$'s, so

$$\mathbf{V}^\top = \begin{bmatrix} \text{---} & \mathbf{v}_1^\top & \text{---} \\ \text{---} & \mathbf{v}_2^\top & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{v}_m^\top & \text{---} \end{bmatrix}.$$

For (20.6.1), this says $\mathbf{V}^\top = \begin{bmatrix} 3 & -2 & 1 \\ 0 & 1 & 4 \end{bmatrix}$. Since the $\mathbf{v}_i$'s are linearly independent, the $m \times m$ matrix $\mathbf{V}^\top\mathbf{V}$ is *always invertible* due to the following general result (and such invertibility is seen by inspection in our running numerical example, for which $\mathbf{V}^\top\mathbf{V}$ is the 2×2 matrix $\begin{bmatrix} 14 & 2 \\ 2 & 17 \end{bmatrix}$):

Theorem 20.6.1. Let A be an $n \times m$ matrix. The $m \times m$ matrix $A^\top A$ is invertible precisely when the columns of A are linearly independent.

A proof of Theorem 20.6.1 occupies the second half of Section 20.8.

Example 20.6.2. To illustrate Theorem 20.6.1 with 3×2 matrices, consider

$$A = \begin{bmatrix} 1 & -3 \\ 4 & 1 \\ -2 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & -3 \\ 4 & -12 \\ -2 & 6 \end{bmatrix}.$$

The columns of A are linearly independent (neither is a scalar multiple of the other) but the columns of B are linearly dependent (the second column is -3 times the first column). By direct calculation

$$A^\top A = \begin{bmatrix} 1 & 4 & -2 \\ -3 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & -3 \\ 4 & 1 \\ -2 & 2 \end{bmatrix} = \begin{bmatrix} 21 & -3 \\ -3 & 14 \end{bmatrix},$$

and this has an inverse: $\begin{bmatrix} 14/285 & 3/285 \\ 3/285 & 21/285 \end{bmatrix}$. In contrast,

$$B^\top B = \begin{bmatrix} 1 & 4 & -2 \\ -3 & -12 & 6 \end{bmatrix} \begin{bmatrix} 1 & -3 \\ 4 & -12 \\ -2 & 6 \end{bmatrix} = \begin{bmatrix} 21 & -63 \\ -63 & 189 \end{bmatrix}$$

is *not* invertible (ultimately because its columns are linearly dependent: the second is -3 times the first, so the system of equations $(B^\top B)\mathbf{x} = \mathbf{0}$ has a solution beyond just $\mathbf{x} = \mathbf{0}$, such as $\mathbf{x} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$). ■

Having introduced the ingredients, here is a compact exact formula via matrix algebra to compute projections and so to solve least-squares problems such as best-fit lines in a new way. (For those who are interested, a proof of the following result is given in Section 20.8.)

Theorem 20.6.3. For linearly independent $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbf{R}^n$ spanning a subspace W of $\mathbf{R}^n$ and the associated $n \times m$ matrix $\mathbf{V}$ whose jth column is $\mathbf{v}_j$, the projection of any $\mathbf{x} \in \mathbf{R}^n$ into W is $\mathbf{Proj}_W\,\mathbf{x} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_m\mathbf{v}_m$ where the coefficients $c_1, \ldots, c_m$ are the entries in the m-vector $\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} = (\mathbf{V}^\top\mathbf{V})^{-1}\mathbf{V}^\top\mathbf{x}$.

Remark 20.6.4. As promised, the formula for $\mathbf{Proj}_W\,\mathbf{x}$ in this result works with *any* basis of W, not just an orthogonal basis. In this respect, it is more convenient for some applications than the formula in Theorem 6.2.1. However, it must be emphasized that the formula in Theorem 6.2.1 is also useful in many practical applications where the notion of orthogonality plays a central role in mathematical models (such as in Fourier analysis for signal processing and in quantum mechanics).

Example 20.6.5. In the running numerical example, $\mathbf{V}^\top\mathbf{V} = \begin{bmatrix} 14 & 2 \\ 2 & 17 \end{bmatrix}$ and $\mathbf{V}^\top\mathbf{x} = \begin{bmatrix} -4 \\ 17 \end{bmatrix}$. Hence, Theorem 20.6.3 says $\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 14 & 2 \\ 2 & 17 \end{bmatrix}^{-1} \begin{bmatrix} -4 \\ 17 \end{bmatrix} = \begin{bmatrix} 17/234 & -2/234 \\ -2/234 & 14/234 \end{bmatrix} \begin{bmatrix} -4 \\ 17 \end{bmatrix} = \begin{bmatrix} -17/39 \\ 41/39 \end{bmatrix}$. Thus, in this example, the projection of $\mathbf{x} = \begin{bmatrix} -2 \\ 1 \\ 4 \end{bmatrix}$ into the plane $\mathrm{span}(\mathbf{v}_1, \mathbf{v}_2)$ is

$$c_1\mathbf{v}_1 + c_2\mathbf{v}_2 = \frac{-17}{39}\mathbf{v}_1 + \frac{41}{39}\mathbf{v}_2 = \begin{bmatrix} -17/13 \\ 25/13 \\ 49/13 \end{bmatrix} \approx \begin{bmatrix} -1.3077 \\ 1.9231 \\ 3.7692 \end{bmatrix}.$$

■

The general formula in Theorem 20.6.3 is convenient in the sense that we can easily apply it on a computer. It may look easier to use than the formula in Theorem 6.2.1 whose applicability requires first finding an orthogonal basis for the linear subspace, a task we saw how to solve in Chapter 19 using the Gram–Schmidt process. However, this apparent ease is a bit of an illusion because the Gram–Schmidt process is lurking in the shadows of applying the formula in Theorem 20.6.3 on a computer: it is hiding in the use of matrix inverses because in many (but not all) settings the way a computer inverts a matrix rests on a procedure called the QR-decomposition (to be discussed in Chapter 22) that is a repackaging of the Gram–Schmidt process.

20.7. Application III: reflections. A class of orthogonal transformations is given by the following construction. For a nonzero vector $\mathbf{v} \in \mathbf{R}^n$, the *reflection through* $\mathbf{v}$ is the linear transformation $r_{\mathbf{v}} : \mathbf{R}^n \to \mathbf{R}^n$ defined by

$$r_{\mathbf{v}}(\mathbf{x}) = \mathbf{x} - 2\left(\frac{\mathbf{x}\cdot\mathbf{v}}{\mathbf{v}\cdot\mathbf{v}}\right)\mathbf{v} = \mathbf{x} - 2\,\mathbf{Proj}_{\mathbf{v}}(\mathbf{x}); \tag{20.7.1}$$

this is linear since $\mathbf{Proj}_{\mathbf{v}}$ is linear.

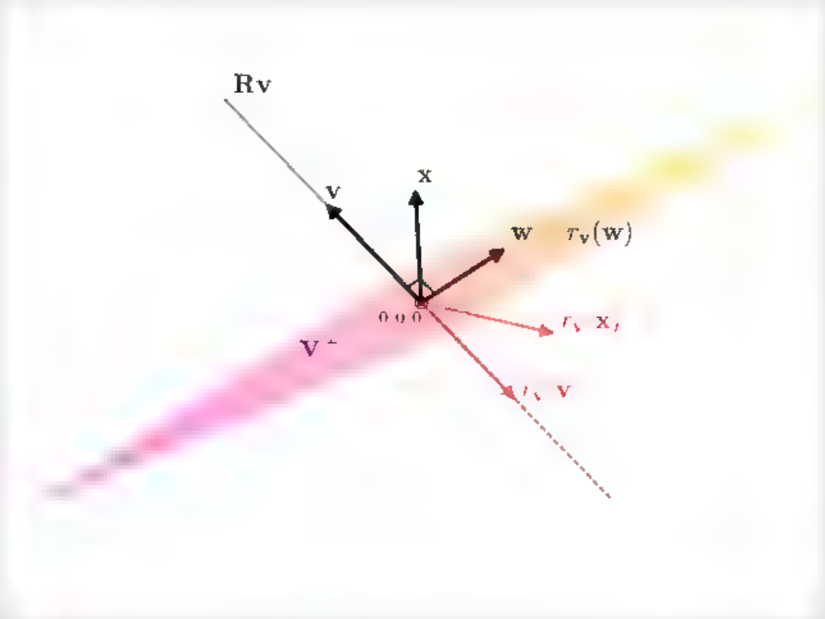

$\mathbf{R}\mathbf{v}$
$\mathbf{v}$
$\mathbf{x}$
$\mathbf{w}$
$r_{\mathbf{v}}(\mathbf{w})$
0 0 0
$\mathbf{v}^{\perp}$
$r_{\mathbf{v}}(\mathbf{x})$
$r_{\mathbf{v}}(\mathbf{v})$

FIGURE 20.7.1. Reflection through a nonzero $\mathbf{v}$ leaves $\mathbf{v}^{\perp}$ unaffected and its effect on the line $\mathrm{span}(\mathbf{v}) = \mathbf{R}\mathbf{v}$ is multiplication by -1 (flipping this line around the origin).

The reason for this name is that (20.7.1) has the geometric effect one wants: the subspace $\mathbf{v}^{\perp}$ of vectors perpendicular to the line $\mathbf{R}\mathbf{v} = \mathrm{span}(\mathbf{v})$ through $\mathbf{v}$ is unaffected by $r_{\mathbf{v}}$ (since $\mathbf{x}\cdot\mathbf{v} = 0$ for such $\mathbf{x}$), whereas on the line $\mathbf{R}\mathbf{v}$ the effect is negation since

$$r_{\mathbf{v}}(c\mathbf{v}) = c\mathbf{v} - 2\frac{c\mathbf{v}\cdot\mathbf{v}}{\mathbf{v}\cdot\mathbf{v}}\mathbf{v} = c\mathbf{v} - 2c\mathbf{v} = -c\mathbf{v},$$

for all $c \in \mathbf{R}$. Thus, $r_{\mathbf{v}}$ looks like reflection across the (hyper)plane $\mathbf{v}^{\perp}$, as is in Figure 20.7.1.

(Many applied references say “Householder transformation” rather than “reflection” because the numerical analyst A.S. Householder did pioneering work [**Hou**] in the late 1950’s using reflections to improve numerical stability for computing the QR-decomposition discussed in Chapter 22. Reflections have been used by pure mathematicians since the early 20th century [**Bou**, Historical Note].)

Example 20.7.1. If $\mathbf{v} = \begin{bmatrix} 3 \\ -2 \\ 2 \end{bmatrix}$ then

$$r_{\mathbf{v}}(x, y, z) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - 2\frac{3x - 2y + 2z}{17}\begin{bmatrix} 3 \\ -2 \\ 2 \end{bmatrix} = \begin{bmatrix} (-x + 12y - 12z)/17 \\ (12x + 9y + 8z)/17 \\ (-12x + 8y + 9z)/17 \end{bmatrix}$$
$$= \begin{bmatrix} -1/17 & 12/17 & -12/17 \\ 12/17 & 9/17 & 8/17 \\ -12/17 & 8/17 & 9/17 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

As this shows, even though $r_{\mathbf{v}}$ has nice geometric meaning, typically its explicit matrix is a real mess (except in special cases such as $\mathbf{v}$ along the direction of a standard basis vector for $\mathbf{R}^n$). ■

The visualization as “reflection through a (hyper)plane” suggests that $r_{\mathbf{v}}$ is length-preserving and hence orthogonal, and this can be verified in general via a calculation of dot products and in specific cases via direct work with matrices. For instance, in Example 20.7.1 the orthogonality of the matrix

$$R = \begin{bmatrix} -1/17 & 12/17 & -12/17 \\ 12/17 & 9/17 & 8/17 \\ -12/17 & 8/17 & 9/17 \end{bmatrix}$$

can be verified by hand (to check the columns are orthonormal) via a tedious calculation. Curiously, the argument in general with dot products *avoiding* explicit matrices is cleaner and more efficient:

$$\begin{aligned} r_{\mathbf{v}}(\mathbf{x}) \cdot r_{\mathbf{v}}(\mathbf{x}) &= (\mathbf{x} - 2\frac{\mathbf{x} \cdot \mathbf{v}}{\mathbf{v} \cdot \mathbf{v}}\mathbf{v}) \cdot (\mathbf{x} - 2\frac{\mathbf{x} \cdot \mathbf{v}}{\mathbf{v} \cdot \mathbf{v}}\mathbf{v}) \\ &= \mathbf{x} \cdot \mathbf{x} - 2\frac{(\mathbf{x} \cdot \mathbf{v})^2}{\mathbf{v} \cdot \mathbf{v}} - 2\frac{(\mathbf{x} \cdot \mathbf{v})(\mathbf{v} \cdot \mathbf{x})}{\mathbf{v} \cdot \mathbf{v}} + 4\frac{(\mathbf{x} \cdot \mathbf{v})^2}{(\mathbf{v} \cdot \mathbf{v})^2}(\mathbf{v} \cdot \mathbf{v}) \\ &= \mathbf{x} \cdot \mathbf{x} - 4\frac{(\mathbf{x} \cdot \mathbf{v})^2}{\mathbf{v} \cdot \mathbf{v}} + 4\frac{(\mathbf{x} \cdot \mathbf{v})^2}{(\mathbf{v} \cdot \mathbf{v})^2}(\mathbf{v} \cdot \mathbf{v}) \\ &= \mathbf{x} \cdot \mathbf{x}. \end{aligned}$$

This preservation of the dot product expresses length-preservation, so $r_{\mathbf{v}} : \mathbf{R}^n \to \mathbf{R}^n$ is orthogonal.

Remark 20.7.2. In Example 20.7.1, the matrix for $r_{\mathbf{v}}$ is not only orthogonal but also *symmetric*. We now explain this geometrically as a feature of the matrix associated with *any* reflection (and an algebraic proof can be given using Proposition 20.3.10, since $r_{\mathbf{v}} = \mathrm{I}_n - 2\,\mathbf{Proj}_{\mathbf{v}}$).

The inverse of an orthogonal $n \times n$ matrix R is given by its transpose $R^\top$, and symmetry is the condition $R^\top = R$, so symmetry for an orthogonal matrix R amounts to the condition that the inverse $R^{-1} = R^\top$ of R is equal to R, which is to say $R^2 = \mathrm{I}_n$. In the case of R corresponding to a reflection $r_{\mathbf{v}}$, the symmetry of R as expressed by the matrix equation $R^2 = \mathrm{I}_n$ says exactly that the composition $r_{\mathbf{v}} \circ r_{\mathbf{v}} : \mathbf{R}^n \to \mathbf{R}^n$ carries each $\mathbf{x}$ to itself. In other words, symmetry of the matrix for $r_{\mathbf{v}}$ says exactly that if we reflect twice through the same (hyper)plane then we wind up exactly where we began.

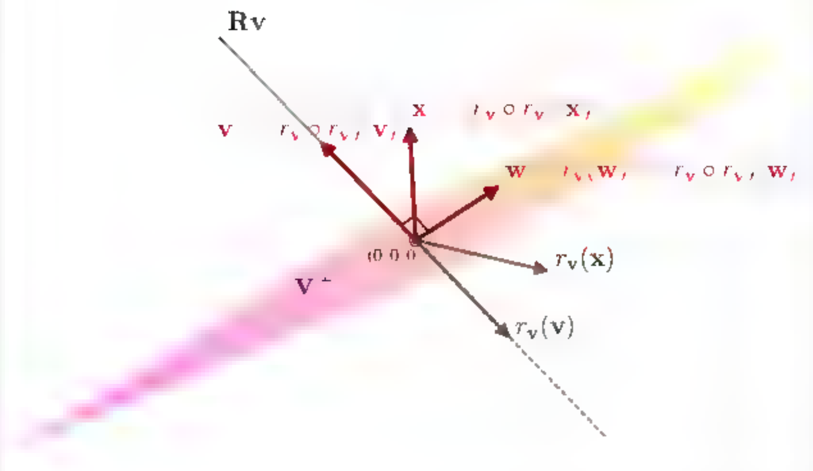

$\mathbf{R}\mathbf{v}$
$\mathbf{v}$ $(r_{\mathbf{v}} \circ r_{\mathbf{v}})(\mathbf{v})$
$\mathbf{x}$ $(r_{\mathbf{v}} \circ r_{\mathbf{v}})(\mathbf{x})$
$\mathbf{w}$ $r_{\mathbf{v}}(\mathbf{w})$ $(r_{\mathbf{v}} \circ r_{\mathbf{v}})(\mathbf{w})$
$(0,0,0)$
$V^{\perp}$
$r_{\mathbf{v}}(\mathbf{x})$
$r_{\mathbf{v}}(\mathbf{v})$

FIGURE 20.7.2. Reflecting through $\mathbf{v}$ twice brings $\mathbf{x}$ back to itself since $r_{\mathbf{v}}(\mathbf{x})$ reflects to $\mathbf{x}$

This is a plausible-sounding property in general due to the visualization for $n = 3$ as in Figure 20.7.2, and in general (for any n) it can be verified by a direct calculation using the linearity of $r_{\mathbf{v}}$:

$$r_{\mathbf{v}}(r_{\mathbf{v}}(\mathbf{x})) = r_{\mathbf{v}}\left(\mathbf{x} - 2\left(\frac{\mathbf{x}\cdot\mathbf{v}}{\mathbf{v}\cdot\mathbf{v}}\right)\mathbf{v}\right) = r_{\mathbf{v}}(\mathbf{x}) - 2\left(\frac{\mathbf{x}\cdot\mathbf{v}}{\mathbf{v}\cdot\mathbf{v}}\right) r_{\mathbf{v}}(\mathbf{v}) = r_{\mathbf{v}}(\mathbf{x}) + 2\left(\frac{\mathbf{x}\cdot\mathbf{v}}{\mathbf{v}\cdot\mathbf{v}}\right)\mathbf{v} = \mathbf{x},$$

where the third equality uses "$r_{\mathbf{v}}(\mathbf{v}) = -\mathbf{v}$" and the final one expresses the definition of $r_{\mathbf{v}}(\mathbf{x})$.

Remark 20.7.3. The fact that rotations around the origin in $\mathbf{R}^3$ are given by orthogonal 3×3 matrices can be refined in a crucial way. All rotations have the additional property of being "orientation-preserving" (which loosely means that they preserve one's sense of how the directions up/down, left/right, and forward/backward relate to each other, and cannot ever convert a physical object into its mirror image). Not only is every rotation around the origin given by an orientation-preserving orthogonal 3×3 matrix, but it works in reverse: the effect on $\mathbf{R}^3$ of every orientation-preserving orthogonal 3×3 matrix is always a rotation around some line through the origin! The equivalence between "rotation around the origin in $\mathbf{R}^3$" and "orientation-preserving orthogonal 3×3 matrix", and the further remarkable fact that all such motions are rotations around a line through the origin, is proved in Remark E.5.3. This is a beautiful link between the geometry of space and the algebra of matrices.

An interesting consequence is the non-obvious fact that the composition of rotations around two lines through the origin in $\mathbf{R}^3$ is a rotation around a line through the origin. Indeed, the composition is certainly orientation-preserving, and it is given by an orthogonal matrix (since a product of any two orthogonal matrices is an orthogonal matrix), so it is a rotation around a line through the origin by the results just mentioned.

20.8. Explanation of the matrix algebra formula for a projection. Where does Theorem 20.6.3 come from? We shall give a general argument, illustrating each step in terms of what it says for the special case in (20.6.1) (with $\mathbf{Y}$ there in the role of $\mathbf{x}$ in Theorem 20.6.3). This rests on Theorem 20.6.1, so we also prove that.

Let $\mathbf{x}^*$ denote $\mathbf{Proj}_W\mathbf{x}$. A key property of $\mathbf{Proj}_W$, established in Section 6.2, is that $\mathbf{x}^* - \mathbf{x}$ is perpendicular to every vector in $W = \mathrm{span}(\mathbf{v}_1, \ldots, \mathbf{v}_m)$. The perpendicularity of $\mathbf{x}^* - \mathbf{x}$ to each $\mathbf{v}_i$

amounts to the conditions $\mathbf{v}_i \cdot (\mathbf{x}^* - \mathbf{x}) = 0$ for every i, which can be equivalently expressed as

$$\mathbf{v}_i \cdot \mathbf{x}^* = \mathbf{v}_i \cdot \mathbf{x} \text{ for every } i. \tag{20.8.1}$$

In the setting of (20.6.1), this amounts to the two conditions

$$\begin{bmatrix} 3 \\ -2 \\ 1 \end{bmatrix} \cdot \mathbf{x}^* = \begin{bmatrix} 3 \\ -2 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} -2 \\ 1 \\ 4 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \\ 4 \end{bmatrix} \cdot \mathbf{x}^* = \begin{bmatrix} 0 \\ 1 \\ 4 \end{bmatrix} \cdot \begin{bmatrix} -2 \\ 1 \\ 4 \end{bmatrix}. \tag{20.8.2}$$

Now we make contact with the features of transpose by rewriting the collection of equations (20.8.1) in matrix notation, using the relationship (20.1.1) between dot products and transposes: the conditions in (20.8.1) say exactly $\mathbf{v}_i^\top \mathbf{x}^* = \mathbf{v}_i^\top \mathbf{x}$ for every i. Assembling these conditions into the entries of a vector, it is the same to assert the single vector equation

$$\begin{bmatrix} \mathbf{v}_1^\top \mathbf{x}^* \\ \mathbf{v}_2^\top \mathbf{x}^* \\ \cdots \\ \mathbf{v}_m^\top \mathbf{x}^* \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1^\top \mathbf{x} \\ \mathbf{v}_2^\top \mathbf{x} \\ \cdots \\ \mathbf{v}_m^\top \mathbf{x} \end{bmatrix} \tag{20.8.3}$$

in $\mathbf{R}^m$. In (20.6.1), this is an equality of 2-vectors: $\begin{bmatrix} [3\ -2\ 1]\mathbf{x}^* \\ [0\ 1\ 4]\mathbf{x}^* \end{bmatrix} = \begin{bmatrix} [3\ -2\ 1]\mathbf{x} \\ [0\ 1\ 4]\mathbf{x} \end{bmatrix}$.

We shall now use $\mathbf{V}$ to rewrite (20.8.3) in a more useful form. Recall that for any column vector $\mathbf{b}$ viewed as an $m \times 1$ matrix and for any $n \times m$ matrix A, the matrix product $A\mathbf{b}$ is the same as a matrix-vector product. In other words, $A\mathbf{b}$ is an $n \times 1$ matrix whose entries are the dot products of the rows of A against the vector $\mathbf{b}$. Applying this observation with $\mathbf{b}$ equal to either $\mathbf{x}$ or $\mathbf{x}^*$, the equality of vectors in (20.8.3) amounts to precisely the matrix equation

$$\mathbf{V}^\top \mathbf{x}^* = \mathbf{V}^\top \mathbf{x} \tag{20.8.4}$$

since the rows of $\mathbf{V}^\top$ are precisely the $\mathbf{v}_i^\top$'s (i.e., the $\mathbf{v}_i$'s as rows). In the running numerical example, this becomes an explicit condition on $\mathbf{x}^*$ that is just a repackaging of (20.8.2):

$$\begin{bmatrix} 3 & -2 & 1 \\ 0 & 1 & 4 \end{bmatrix} \mathbf{x}^* = \begin{bmatrix} 3 & -2 & 1 \\ 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} -2 \\ 1 \\ 4 \end{bmatrix}. \tag{20.8.5}$$

The *unknown* $\mathbf{x}^*$ belongs to $\mathrm{span}(\mathbf{v}_1, \dots, \mathbf{v}_m)$, so $\mathbf{x}^* = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_m\mathbf{v}_m$ for some coefficients $c_1, \dots, c_m$ that we *do not yet know*. In matrix language, this says $\mathbf{x}^* = \mathbf{V} \cdot \begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix}$. Plugging that into the left side of (20.8.4) yields

$$\mathbf{V}^\top \mathbf{V} \begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix} = \mathbf{V}^\top \mathbf{x}. \tag{20.8.6}$$

In the running numerical example, this says:

$$\begin{bmatrix} 3 & -2 & 1 \\ 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ -2 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 3 & -2 & 1 \\ 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} -2 \\ 1 \\ 4 \end{bmatrix},$$

or equivalently (upon multiplying out the numerical matrix products)

$$\begin{bmatrix} 14 & 2 \\ 2 & 17 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} -4 \\ 17 \end{bmatrix}. \tag{20.8.7}$$

Provided that the matrix $\mathbf{V}^\top\mathbf{V}$ is *invertible*, we can multiply both sides of (20.8.6) by $(\mathbf{V}^\top\mathbf{V})^{-1}$ on the left to arrive at the solution

$$\begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix} = (\mathbf{V}^\top\mathbf{V})^{-1}\mathbf{V}^\top\mathbf{x}. \tag{20.8.8}$$

as asserted in Theorem 20.6.3. So it remains to confirm that $\mathbf{V}^\top\mathbf{V}$ is invertible, and since the columns $\mathbf{v}_1, \dots, \mathbf{v}_m$ of $\mathbf{V}$ are assumed to be linearly independent such invertibility is exactly the content of Theorem 20.6.1. So to complete the proof of Theorem 20.6.3, it remains to prove Theorem 20.6.1.

Proof of Theorem 20.6.1. Letting A now be as in Theorem 20.6.1, with jth column denoted as $\mathbf{a}_j \in \mathbf{R}^n$, the easier part is to show that if the columns $\mathbf{a}_j$ are linearly dependent then $A^\top A$ is *not* invertible. Say we have a linear dependence relation $\sum t_j\mathbf{a}_j = 0$ for some $\mathbf{t} = (t_1, \dots, t_m) \in \mathbf{R}^m$ different from $\mathbf{0}$ (i.e., some t_j is nonzero). But $A\mathbf{t} = \sum t_j\mathbf{a}_j$ and this vanishes by how we chose the t_j's, so $(A^\top A)\mathbf{t} = A^\top(A\mathbf{t}) = A^\top\mathbf{0} = \mathbf{0}$. Hence, $A^\top A$ takes the *different* inputs $\mathbf{t}$ and $\mathbf{0}$ to the same output $\mathbf{0}$, so $A^\top A$ is not invertible.

Now suppose the columns $\mathbf{a}_j$ of A are linearly independent. We want to show that $A^\top A$ is invertible, or in other words that $A^\top A\mathbf{x} = \mathbf{b}$ has a unique solution $\mathbf{x} \in \mathbf{R}^m$ for any $\mathbf{b} \in \mathbf{R}^m$. First we check uniqueness: if $A^\top A\mathbf{x}' = \mathbf{b} = A^\top A\mathbf{x}$ for some $\mathbf{x}, \mathbf{x}' \in \mathbf{R}^m$ then we claim that $\mathbf{x}' = \mathbf{x}$. We have

$$A^\top A(\mathbf{x}' - \mathbf{x}) = A^\top A\mathbf{x}' - A^\top A\mathbf{x} = \mathbf{b} - \mathbf{b} = \mathbf{0},$$

and our aim is to show that $\mathbf{x}' = \mathbf{x}$ or equivalently that $\mathbf{x}' - \mathbf{x} = \mathbf{0}$. Thus, the uniqueness aspect comes down to showing that if $A^\top A\mathbf{v} = \mathbf{0}$ then $\mathbf{v} = \mathbf{0}$. For this we shall use the relationship of transpose with dot products!

The vanishing of $A^\top A\mathbf{v}$ gives (with help from multiplicative properties of transpose) that

$$0 = \mathbf{v} \cdot (A^\top A\mathbf{v}) = \mathbf{v}^\top(A^\top A\mathbf{v}) = (\mathbf{v}^\top A^\top)(A\mathbf{v}) = (A\mathbf{v})^\top(A\mathbf{v}) = A\mathbf{v} \cdot A\mathbf{v} = \|A\mathbf{v}\|^2,$$

so $A\mathbf{v} = \mathbf{0}$. But if $v_1, \dots, v_n$ are the entries in $\mathbf{v} \in \mathbf{R}^m$ then $A\mathbf{v} = \sum v_j\mathbf{a}_j$, so the vanishing of $A\mathbf{v}$ says $\sum v_j\mathbf{a}_j = \mathbf{0}$. By hypothesis the $\mathbf{a}_j$'s are linearly independent, so this forces all v_j's to vanish, and hence $\mathbf{v} = \mathbf{0}$ as desired.

For the $m \times m$ matrix $M = A^\top A$ we have shown that the linear system $M\mathbf{x} = \mathbf{b}$ of m equations in m unknowns has *at most one* solution for any given $\mathbf{b} \in \mathbf{R}^m$. We need to show that it actually has a solution. The key observation is that the matrix-vector product $M\mathbf{x} \in \mathbf{R}^m$ is the linear combination of the m columns of M in which the ith column is multiplied against the ith entry of $\mathbf{x}$: if we write $\mathbf{v}_1, \dots, \mathbf{v}_m$ for the respective columns of M then we are saying that $M\mathbf{x} = \sum x_i\mathbf{v}_i$. This assertion for a general $m \times m$ matrix M is just a matter of the explicit definition of matrix-vector products, and is perhaps most readily grasped by working it out for $m = 2$ (the case of general m being the same except for requiring more notation, and is illustrated in an example with $m = 3$ in (21.2.3)):

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix} = x\begin{bmatrix} a \\ c \end{bmatrix} + y\begin{bmatrix} b \\ d \end{bmatrix}.$$

Hence, we can reformulate our task in geometric terms: we want to show that the columns $\mathbf{v}_1, \dots, \mathbf{v}_m$ span $\mathbf{R}^m$. Letting V denote the span of these columns, by Theorem 4.2.8 applied to the containment of V in $\mathbf{R}^m$ we have $\dim V \leq m$ with equality precisely when $V = \mathbf{R}^m$.

The fact that $M\mathbf{x} = \mathbf{0}$ has as its *only* solution $\mathbf{x} = \mathbf{0}$ is precisely the assertion that the *only* scalars $x_1, \ldots, x_m$ for which $\sum x_i \mathbf{v}_i = \mathbf{0}$ are $x_1 = 0, \ldots, x_m = 0$. In other words, the $\mathbf{v}_i$'s are linearly independent. But there are m of these $\mathbf{v}_i$'s, so their span has dimension m by Theorem 19.2.3. $\square$

Chapter 20 highlights (links to highlights in and next chapters)

Notation	**Meaning**	**Location in text**
$A^\top$ for $m \times n$ matrix A	transpose of A (it is an $n \times m$ matrix)	Definition 20.1.1
$q_A(\mathbf{x})$ for $n \times n$ symmetric matrix A	for $\mathbf{x} \in \mathbf{R}^n$, $q_A(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x} \in \mathbf{R}$	(20.3.1)

Concept	**Meaning**	**Location in text**
transpose of $m \times n$ matrix A	the $n \times m$ matrix obtained by flipping A across its diagonal	Definition 20.1.1
quadratic form in n variables	expression in $x_1, \dots, x_n$ involving sum of terms $c_{ij}x_ix_j$ (with $i \leq j$)	Definition 20.3.1, Example 20.3.3
symmetric $n \times n$ matrix	equal ij-entry and ji-entry for all i, j	Definition 20.3.5
orthogonal $n \times n$ matrix A	$T_A : \mathbf{R}^n \to \mathbf{R}^n$ is length-preserving (same as columns being orthonormal)	Definition 20.4.2

Result	**Meaning**	**Location in text**
transpose expresses how $m \times n$ matrix A moves across a dot product	$(A\mathbf{x}) \cdot \mathbf{y} = \mathbf{x} \cdot (A^\top\mathbf{y})$ for $\mathbf{x} \in \mathbf{R}^n$, $\mathbf{y} \in \mathbf{R}^m$	Theorem 20.1.4
powers and inverse of symmetric $n \times n$ matrix A are symmetric	A^r symmetric for all $r \geq 1$, A^{-1} symmetric if A invertible	Exs. 20.3.6, 20.3.7
Gram matrices and projection matrices are symmetric	$M^\top M$ symmetric for $m \times n$ matrix M, for linear subspace V of $\mathbf{R}^n$ the $n \times n$ matrix for $\mathbf{Proj}_V : \mathbf{R}^n \to \mathbf{R}^n$ is symmetric	Thm. 20.3.8, Prop. 20.3.10
characterizations of orthogonal $n \times n$ matrices	list of equivalent geometric and algebraic conditions that define orthogonality for an $n \times n$ matrix	Theorem 20.4.1
relation of orthogonal matrices to inversion and products	A^{-1} is orthogonal if A is, AB orthogonal if A and B are	Proposition 20.4.4

Skill	**Location in text**
relate transpose to multiplication (flips order!) and inversion, and rewrite dot product as matrix product via transpose	box containing (20.1.1), Examples 20.1.7–20.1.9
determine by inspection if a matrix is symmetric	Example 20.3.4
go both ways through the dictionary between n-variable quadratic forms and $n \times n$ symmetric matrices (especially with $n = 2, 3$)	Example 20.3.11–20.3.12 and box containing (20.3.2) and (20.3.3)
determine by inspection of columns if a square matrix is orthogonal	orthonormality criterion in Theorem 20.4.1

21. Linear systems, column space, and null space

At several points in this book, we have encountered systems of simultaneous linear equations, which are often referred to as *linear systems*. To be completely specific, a "system of simultaneous linear equations" is a collection of linear equations in some unknowns $x_1, \dots, x_n$. For example, we might seek all $x, y, z \in \mathbf{R}$ that satisfy all three equations

$$2x + y - z = 1, \quad x + 2y + z = 4, \quad -x - y + z = 2.$$

We have learned to express this more succinctly in matrix notation: find all $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbf{R}^3$ for which

$$\begin{bmatrix} 2 & 1 & -1 \\ 1 & 2 & 1 \\ -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix}.$$

A general *linear system* (of m equations in n unknowns) takes the form

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n &= b_m, \end{aligned}$$

where the a_{ij}'s and b_i's are given constants; this can also be written in matrix form $A\mathbf{x} = \mathbf{b}$ with an $m \times n$ matrix A, an m-vector $\mathbf{b}$, and an unknown vector $\mathbf{x} \in \mathbf{R}^n$.

It would not be a stretch to say that the study of linear systems is the main problem of linear algebra. Of course, as you have already seen, linear algebra goes in many directions and has a huge variety of applications, many of which do not look like they involve linear systems. However, a closer inspection shows that at the heart of all of these situations is some linear system. Many textbooks take linear systems as the primary perspective for developing linear algebra. This book has chosen a different and broader path, but it is important to recognize the centrality of linear systems.

The use of matrix notation to express a linear system is not only substantially more concise than writing out the equations in full, but also suggests algebraic manipulations that might not occur to us otherwise. For example, if the matrix A associated with some linear system is known to be *invertible* (so $m = n$) then we learned in Section 18.3 that this system of equations can be solved by multiplying both sides by the inverse matrix, leading to the unique solution $\mathbf{x} = A^{-1}\mathbf{b}$. This holds for the linear system in (x, y, z) at the start above, which has the unique solution $\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2 & 1 & -1 \\ 1 & 2 & 1 \\ -1 & -1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix}$.

There remains the fundamental issue (when $m = n$) of determining when a given $n \times n$ matrix *does* have an inverse, and if so then *actually computing* the inverse matrix. (For the case above with $n = 3$, upon computing the 3×3 inverse matrix one obtains the unique solution $(x, y, z) = (3, -4/3, 11/3)$.) Furthermore, for many linear systems that arise in practice, the matrix A is *not* invertible. This happens whenever A is not a square matrix; i.e., whenever $m \neq n$. What do we do in such cases? Our goal in this chapter is to develop some of the tools needed to answer questions such as:

(i) When does a given linear system $A\mathbf{x} = \mathbf{b}$ of m equations in n unknowns have a solution?
(ii) If there is a solution, is there more than one? How do we describe the set of all solutions?
(iii) Are there sometimes ways to "easily" determine if there is a solution or not?

Ideally we would like practical algorithmic answers to these questions. In other words, we seek systematic and efficiently computable algorithms, even when m and n are very large, that not only answer these questions but also give us the answer in a useful way. We are going to answer (i) and partially answer (ii) in this chapter, and say more about (ii) and give an approach to (iii) in Chapter 22.

By the end of this chapter, you should be able to:

- define the column space and null space of any matrix;
- compute an orthogonal basis of the column space of a general matrix A and use it to determine if $A\mathbf{x} = \mathbf{b}$ has a solution, and compute a basis for the null space for a 2×2 matrix;
- know how to recognize overdetermined and underdetermined linear systems, and draw correct qualitative conclusions (e.g., 0, 1, or infinitely many solutions).

21.1. Examples of linear systems. Before proceeding to answer some of the questions posed above, we provide a handful of examples to illustrate that systems of simultaneous linear equations in many variables (i.e., where n is much larger than 3) arise in many natural and important situations that you are sure to encounter no matter what you go on to study.

(1) Balancing chemical reactions and balancing forces in a mechanical system lead to linear systems. The structural analysis of an arena roof, oil rig, or tall building via the "finite element method" can involve linear systems in *thousands* of variables. Balancing problems in chemistry are intractable without linear algebra if there are more than a few atoms. (See [BV, 8.3.1, 11.4] for examples with chemical reactions, including conservation of charge.)

(2) In data science, finance, and signal processing one often works with $\mathbf{R}^n$ for very large n (e.g., an electrical signal expressed via many sinusoidal waves with varying amplitudes and phases, managing a portfolio of many investments, or measurements made at many times for high-frequency trading). Extracting useful information from a vector formulation of a problem (to fit parameters in a financial model, rank webpages, denoise a signal, etc.) rests on ideas and methods of linear algebra: artful manipulation of large linear systems and finding solutions or approximate solutions with desired properties (see Example 22.5.3).

(3) Kirchhoff's laws lead to a linear system relating the currents in different parts of a "linear" circuit (see Example 14.2.3); this is an electrical analogue of balancing forces in a mechanical system. For modern circuits, these linear systems can involve more than 100,000 variables! (See [M, Sec. 2.6, Ex. 4.4.6–4.4.7] for how to recast *every* such circuit problem in terms of linear algebra and a proof via the techniques in this chapter that all such problems have a unique solution. Thus, Kirchhoff's laws are a "complete" set of laws for circuits.)

(4) Determining the 3-dimensional structure of a protein from nuclear magnetic resonance (NMR) or X-ray crystallography measurements leads to a large linear system (due to the large number of atoms involved).

(5) Very similar to (3), medical imaging (e.g., reconstructing images from the measurements taken by CAT scans or MRI's) is based on repeatedly solving very large linear systems.

(6) The input-output model for studying interdependencies among various components of an economy (for which Leontief won the 1973 Nobel Prize in economics) amounts to formulating a specific linear system and studying its solution(s). The novel feature of this model, which was developed in the late 1930's and early 1940's, is that the interdependencies create a feedback loop within the equations. This enables one to use techniques based on matrix powers to (i) show that under economically reasonable assumptions there is always exactly one solution, and (ii) rapidly approximate it. (See [M, Ex. 8.3.6] for a discussion of this

model and how it is analyzed using ideas related to those underlying why the PageRank algorithm works, and see [MB, Ch. 2] for an example-rich treatment of the model.)

(7) Many real-life calculus problems (e.g., climate modeling, analysis of structures in mechanical engineering, wave propagation, etc.) are far too complicated to solve exactly; instead one approximates them with a suitable linear system, often involving thousands or millions of variables.

21.2. Column space. We now begin the systematic study of linear systems. Let's write a system of m linear equations in n variables $x_1, \dots, x_n$ in matrix language as $A\mathbf{x} = \mathbf{b}$ where the $m \times n$ matrix A encodes the coefficients and the m-vector $\mathbf{b}$ encodes the constants on the right side of the equations. A bit more explicitly, this says:

$$A \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}. \qquad (21.2.1)$$

In the introduction to this chapter, we posed some fundamental questions concerning (21.2.1). We begin with the first of these:

(i) Given A, for which $\mathbf{b} \in \mathbf{R}^m$ does $A\mathbf{x} = \mathbf{b}$ have a solution $\mathbf{x} \in \mathbf{R}^n$?

Let us first revisit the example at the start of this chapter:

$$\begin{bmatrix} 2 & 1 & -1 \\ 1 & 2 & 1 \\ -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix},$$

so in this case we have

$$A = \begin{bmatrix} 2 & 1 & -1 \\ 1 & 2 & 1 \\ -1 & -1 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix}. \qquad (21.2.2)$$

Observe that the matrix-vector product $A \begin{bmatrix} x \\ y \\ z \end{bmatrix}$ equals

$$A \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2x + y - z \\ x + 2y + z \\ -x - y + z \end{bmatrix} = x \begin{bmatrix} 2 \\ 1 \\ -1 \end{bmatrix} + y \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix} + z \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}. \qquad (21.2.3)$$

The expression on the left in (21.2.3) is just the initial set of linear equations (or rather, it is the "left side" of those equations), but the right side of (21.2.3) provides a new interpretation: it exhibits the matrix-vector product $A\mathbf{x}$ as a *linear combination of the columns* of the matrix A. This is **extremely important**, so please check to make sure you see why the right side of (21.2.3) is correct.

We can now reinterpret the original algebraic problem of solving 3 equations in 3 unknowns in geometric terms: it is equivalent to ask whether the vector $\mathbf{b}$ in (21.2.2) lies in the span of the vectors given by the columns of the 3×3 matrix A in (21.2.2).

This same insight works in general: if the n columns of the $m \times n$ matrix A in (21.2.1) are denoted $\mathbf{a}_1, \dots, \mathbf{a}_n \in \mathbf{R}^m$ then (just as we saw in Theorem 13.4.1)

$$A \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1\mathbf{a}_1 + \cdots + x_n\mathbf{a}_n. \tag{21.2.4}$$

This leads to a first "answer" to the question we have posed: (21.2.1) has a solution precisely when the vector $\mathbf{b}$ is a linear combination of the columns $\mathbf{a}_1, \dots, \mathbf{a}_n$ of A. In more geometric terms:

A solution exists for $A\mathbf{x} = \mathbf{b}$ precisely when $\mathbf{b}$ lies in the span of the vectors $\mathbf{a}_1, \dots, \mathbf{a}_n \in \mathbf{R}^m$.

The span of these vectors, $\mathbf{a}_1, \dots, \mathbf{a}_n$ (the columns of A) has a special name:

Definition 21.2.1. The *column space* of an $m \times n$ matrix A is the span in $\mathbf{R}^m$ of the columns of A; it is denoted $C(A)$. In other words:

$$C(A) = \text{column space of } A = \text{span in } \mathbf{R}^m \text{ of the columns of } A.$$

The column space $C(A)$ is a linear subspace of $\mathbf{R}^m$ (recall that the columns of A are m-vectors), and for $\mathbf{b} \in \mathbf{R}^m$ the vector equation $A\mathbf{x} = \mathbf{b}$ (a system of m linear equations in n unknowns) has a solution precisely when $\mathbf{b} \in C(A)$.

Please note that all that we have done here is give a linguistic reformulation of the problem. We have not yet explained how you might determine if $\mathbf{b}$ lies in this span. This way of expressing the problem not only gives a way to *visualize* what it means for there to be a solution of this linear system for a given $\mathbf{b}$ but also will lead to ways to *find* the solution (or solutions).

Example 21.2.2. Consider the 2×3 matrix

$$A = \begin{bmatrix} 6 & 1 & 2 \\ 2 & 1 & 3 \end{bmatrix}.$$

By definition, the column space $C(A)$ consists of those vectors $\mathbf{x} \in \mathbf{R}^2$ that are a linear combination of the three 2-vectors $\begin{bmatrix} 6 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \end{bmatrix}$. In particular, $C(A)$ is a linear subspace of $\mathbf{R}^2$. (Make sure you understand why $C(A)$ here is a linear subspace of $\mathbf{R}^2$ rather than $\mathbf{R}^3$.)

Which subspace of $\mathbf{R}^2$ is $C(A)$? There is not much room in $\mathbf{R}^2$ since it is only 2-dimensional. Indeed, since at least one of these column vectors is not the zero vector, $C(A)$ is either a line through the origin or all of $\mathbf{R}^2$. We claim that it cannot be a line, so $C(A)$ must equal $\mathbf{R}^2$.

The only way that $C(A)$ could be a line is if all three columns of A are scalar multiples of one another. (If any two of them 'point in different directions' then their span is not just a line). In this example $\begin{bmatrix} 6 \\ 2 \end{bmatrix}$ lies on the line through the origin with slope $1/3$, while $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ lies on the line through the origin with slope 1 (and the third vector $\begin{bmatrix} 2 \\ 3 \end{bmatrix}$ lies on the line through the origin with slope $3/2$). Another way to say this is that there are no scalars c, c' for which the nonzero columns $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ satisfy $\mathbf{a}_2 = c\mathbf{a}_1$, $\mathbf{a}_3 = c'\mathbf{a}_1$. We have thus shown that $C(A) = \mathbf{R}^2$.

Having shown that $C(A) = \mathbf{R}^2$, we conclude that *every* vector $\mathbf{b} \in \mathbf{R}^2$ lies in $C(A)$; i.e., every $\mathbf{b} \in \mathbf{R}^2$ is a linear combination of the columns of A. This means precisely that $A\mathbf{x} = \mathbf{b}$ has a solution (for *every*

2-vector $\mathbf{b}$). Be careful here: we are *not* saying that there is only one solution to the linear system $A\mathbf{x} = \mathbf{b}$ for each $\mathbf{b}$, only that there is *at least one solution* for each $\mathbf{b}$. We will see later (in Examples 21.3.4 and 21.4.6) that this particular linear system has infinitely many solutions, no matter which $\mathbf{b}$ we choose! ■

We next consider an example whose column space $C(A)$ is *not* the entire space $\mathbf{R}^m$, so there definitely exist choices of $\mathbf{b}$ for which $A\mathbf{x} = \mathbf{b}$ has no solution at all.

Example 21.2.3. The column space of the 3×3 matrix

$$A' = \begin{bmatrix} 2 & 1 & 4 \\ 4 & 2 & 3 \\ 6 & 3 & 1 \end{bmatrix}$$

is the set of all vectors in $\mathbf{R}^3$ that can be written as a linear combination

$$x_1 \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} + x_2 \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + x_3 \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix}. \qquad (21.2.5)$$

In this particular case, the first and second columns of A' are multiples of one another (the first is twice the second), which means that we can combine the first two terms in (21.2.5) to rewrite that linear combination as

$$(2x_1 + x_2) \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + x_3 \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix}.$$

What this says is that any linear combination of the three columns of A' could just as well be written as a linear combination (with some other scalar multiples) of just the second and third columns of A' (or of just the first and third columns). Can we condense it even further? Inspecting these second and third columns, you can see via ratios of corresponding entries that they are not scalar multiples of one another, so these two vectors are linearly independent. Hence, their span (which we have seen is the same as the span of all three columns of A') is a plane through the origin in $\mathbf{R}^3$. That is, $C(A')$ *is 2-dimensional.*

Now suppose $\mathbf{b}$ is a vector in $\mathbf{R}^3$ *not* in this plane. Then it is impossible to write it as a linear combination of the columns of A', so it is impossible to solve $A'\mathbf{x} = \mathbf{b}$. This is one of the central ideas in this chapter (i.e., apply ideas of linear algebra, such as dimension, to the column space), so make sure you understand it! Put another way, the linear system $A'\mathbf{x} = \mathbf{b}$ of 3 equations in 3 unknowns has *no* solution whenever $\mathbf{b}$ does not lie in the plane $C(A')$. On the other hand, this system does have a solution whenever $\mathbf{b}$ does lie in this plane (it will even have infinitely many solutions: see Example 21.3.4). ■

For concreteness, we now state these conclusions for linear systems of 2 equations in 3 unknowns and of 3 equations in 3 unknowns before we go to the general case. First, we introduce some useful notation.

Definition 21.2.4. For any two sets V and W of objects (e.g., linear subspaces of $\mathbf{R}^7$, each considered as a collection of points), the notation "$V \subset W$" read as "V is contained in W" or "V is a subset of W" means that every object of the set V also belongs to the set W. (The case of a line V contained in a plane W is a good one to have in mind.)

The most important instance of this notation for our purposes is when V and W are linear subspaces of $\mathbf{R}^n$, so then "$V \subset W$" means that every vector belonging to V also belongs to W.

Proposition 21.2.5. Let A be a 2×3 matrix whose columns are all nonzero. The subspace $C(A) \subset \mathbf{R}^2$ is a line when all columns are multiples of one another, or equivalently, "have the same slope"; if this is not the case then $C(A) = \mathbf{R}^2$. For any such matrix A and any $\mathbf{b} \in \mathbf{R}^2$, the linear system $A\mathbf{x} = \mathbf{b}$ of 2 equations in 3 unknowns has a solution precisely in the following circumstances:

- if $C(A)$ is a line then there is a solution exactly when $\mathbf{b}$ lies on that line (or more concretely, either $\mathbf{b} = \mathbf{0}$ or the slope b_2/b_1 is the same as that of all nonzero vectors in the line $C(A)$);
- if $C(A) = \mathbf{R}^2$ then there is a solution for any $\mathbf{b}$.

Proposition 21.2.6. Let A be a 3×3 matrix whose columns are all nonzero. The subspace $C(A) \subset \mathbf{R}^3$ is a line when all columns are scalar multiples of each other, it is equal to $\mathbf{R}^3$ when the three columns are linearly independent, and in all other cases it is a plane.

For any such A and any $\mathbf{b} \in \mathbf{R}^3$, the linear system $A\mathbf{x} = \mathbf{b}$ of 3 equations in 3 unknowns has a solution precisely in the following circumstances (depending on $\dim C(A)$):

- if $C(A)$ is a line then there is a solution exactly when $\mathbf{b}$ lies in that line (more concretely, $\mathbf{b}$ is a scalar multiple of any one of the three columns);
- if $C(A)$ is a plane then there is a solution exactly when $\mathbf{b}$ lies in that plane;
- if $C(A) = \mathbf{R}^3$ then there is a solution for any $\mathbf{b}$.

This gives us the first glimmer of an idea for how to determine whether $A\mathbf{x} = \mathbf{b}$ has a solution for a given matrix A and vector $\mathbf{b}$. We will first discuss this for linear systems of 3 equations in 3 unknowns (i.e., $m = n = 3$), and then turn it into a general method (for any m and n). By definition, $C(A)$ is a span of three vectors in $\mathbf{R}^3$. From our experience in Section 5.1 computing the dimension of the span of three vectors in $\mathbf{R}^3$, we can quickly determine which of the cases (in Proposition 21.2.6) we are in.

Even better, applying the Gram–Schmidt process to the collection of 3 columns of A produces an orthogonal basis for their span $C(A)$: the nonzero vectors produced by the Gram–Schmidt process are such a basis, so the number of such nonzero vectors is $\dim C(A)$. For instance, if the Gram–Schmidt process produces only two nonzero vectors $\mathbf{w}$ and $\mathbf{w}'$ then $C(A)$ is a plane. Moreover, having an orthogonal basis for $C(A)$ leads to the following very useful additional feature which will readily adapt to general linear systems below.

Suppose for example that $C(A)$ is 2-dimensional, and Gram–Schmidt has produced an orthogonal basis $\{\mathbf{w}, \mathbf{w}'\}$ for $C(A)$. For any linear subspace W of any $\mathbf{R}^m$, a vector $\mathbf{v} \in \mathbf{R}^m$ belongs to W precisely when $\mathbf{Proj}_W(\mathbf{v}) = \mathbf{v}$ (indeed, the output of $\mathbf{Proj}_W$ always belongs to W, and anything in W is its own closest point in W and so equals its projection into W). Hence, if $\mathbf{b}$ is any vector in $\mathbf{R}^3$ then we can check whether it lies in $C(A)$ by computing $\mathbf{Proj}_{C(A)}(\mathbf{b})$ and checking if this is equal to $\mathbf{b}$ (in which case $A\mathbf{x} = \mathbf{b}$ has a solution) or is not equal to $\mathbf{b}$ (in which case $A\mathbf{x} = \mathbf{b}$ has no solution). The crucial point is that we can *compute* $\mathbf{Proj}_{C(A)}(\mathbf{b})$ since we have in hand an orthogonal basis of $C(A)$ (namely, $\{\mathbf{w}, \mathbf{w}'\}$)! Explicitly,

$$\mathbf{Proj}_{C(A)}(\mathbf{b}) = \mathbf{Proj}_{\mathbf{w}}(\mathbf{b}) + \mathbf{Proj}_{\mathbf{w}'}(\mathbf{b}), \tag{21.2.6}$$

and we check if this is equal to $\mathbf{b}$ or not. Most $\mathbf{b} \in \mathbf{R}^3$ don't lie in the plane $C(A)$, so "usually" the equality (21.2.6) fails and correspondingly for "most" $\mathbf{b}$ the linear system $A\mathbf{x} = \mathbf{b}$ has no solution.

Let us see this in action:

Example 21.2.7. In Example 21.2.3 we saw that for the 3×3 matrix

$$A' = \begin{bmatrix} 2 & 1 & 4 \\ 4 & 2 & 3 \\ 6 & 3 & 1 \end{bmatrix},$$

the column space $C(A') \subset \mathbf{R}^3$ is the plane spanned by second and third columns that are linearly independent (as they are not scalar multiples of each other, by inspection). In other words,

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix} \quad \text{form a basis for } C(A').$$

Using the recipe in Theorem 7.1.1, an orthogonal basis of $C(A')$ is $\{\mathbf{y}, \mathbf{x}'\}$, where $\mathbf{x}' = \mathbf{x} - \mathbf{Proj}_{\mathbf{y}}(\mathbf{x})$; we compute

$$\mathbf{Proj}_{\mathbf{y}}(\mathbf{x}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\mathbf{y} \cdot \mathbf{y}} \mathbf{y} = \frac{13}{26} \mathbf{y} = \frac{1}{2} \mathbf{y},$$

so

$$\mathbf{x}' = \mathbf{x} - \mathbf{Proj}_{\mathbf{y}}(\mathbf{x}) = \mathbf{x} - \frac{1}{2} \mathbf{y} = \begin{bmatrix} -1 \\ 1/2 \\ 5/2 \end{bmatrix}.$$

(As a check on the work, it is a good idea to compute directly that $\mathbf{x}' \cdot \mathbf{y}$ is indeed equal to 0.)

Thus, for any $\mathbf{b} \in \mathbf{R}^3$, the linear system $A'\mathbf{x} = \mathbf{b}$ of 3 equations in 3 unknowns has a solution precisely when

$$\mathbf{b} = \mathbf{Proj}_{\mathbf{y}}(\mathbf{b}) + \mathbf{Proj}_{\mathbf{x}'}(\mathbf{b})$$

where $\mathbf{y}$ and $\mathbf{x}'$ are as above. Let us carry this out for two different $\mathbf{b}$'s:

$$\mathbf{b}_1 = \begin{bmatrix} -3 \\ 4 \\ 13 \end{bmatrix}, \quad \mathbf{b}_2 = \begin{bmatrix} -2 \\ 5 \\ 2 \end{bmatrix}.$$

We first compute the necessary dot products:

$$\mathbf{b}_1 \cdot \mathbf{y} = 13, \quad \mathbf{b}_1 \cdot \mathbf{x}' = \frac{75}{2}, \quad \mathbf{b}_2 \cdot \mathbf{y} = 9, \quad \mathbf{b}_2 \cdot \mathbf{x}' = \frac{19}{2}.$$

Then

$$\begin{aligned} \mathbf{Proj}_{\mathbf{y}}(\mathbf{b}_1) + \mathbf{Proj}_{\mathbf{x}'}(\mathbf{b}_1) = \frac{\mathbf{b}_1 \cdot \mathbf{y}}{\mathbf{y} \cdot \mathbf{y}} \mathbf{y} + \frac{\mathbf{b}_1 \cdot \mathbf{x}'}{\mathbf{x}' \cdot \mathbf{x}'} \mathbf{x}' = \frac{13}{26} \mathbf{y} + \frac{75/2}{15/2} \mathbf{x}' &= \frac{1}{2} \mathbf{y} + 5\,\mathbf{x}' \\ &= \frac{1}{2} \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix} + 5 \begin{bmatrix} -1 \\ 1/2 \\ 5/2 \end{bmatrix} \\ &= \begin{bmatrix} -3 \\ 4 \\ 13 \end{bmatrix}, \end{aligned}$$

which is $\mathbf{b}_1$, so $\mathbf{b}_1 \in C(A')$.

This calculation tells us more: it gives an *actual* solution since the projection calculation exhibits a description of $\mathbf{b}_1$ in terms of the 2nd column $\mathbf{x}$ and 3rd column $\mathbf{y}$ of A'. Indeed, the calculation yields

$$\mathbf{b}_1 = \frac{1}{2} \mathbf{y} + 5\mathbf{x}' = \frac{1}{2} \mathbf{y} + 5(\mathbf{x} - \frac{1}{2} \mathbf{y}) = 5\mathbf{x} - 2\mathbf{y} = A' \begin{bmatrix} 0 \\ 5 \\ -2 \end{bmatrix}, \tag{21.2.7}$$

so $\mathbf{x}_0 = \begin{bmatrix} 0 \\ 5 \\ -2 \end{bmatrix}$ is a solution to $A'\mathbf{x} = \mathbf{b}_1$ (as can be directly verified too). There are many other solutions, such as $\begin{bmatrix} 1 \\ 3 \\ -2 \end{bmatrix}$ and $\begin{bmatrix} -2 \\ 9 \\ -2 \end{bmatrix}$ (see Example 21.3.17 for how to find these and in fact all solutions).

On the other hand,

$$\mathbf{Proj}_{\mathbf{y}}(\mathbf{b}_2) + \mathbf{Proj}_{\mathbf{x}'}(\mathbf{b}_2) = \frac{\mathbf{b}_2 \cdot \mathbf{y}}{\mathbf{y} \cdot \mathbf{y}}\mathbf{y} + \frac{\mathbf{b}_2 \cdot \mathbf{x}'}{\mathbf{x}' \cdot \mathbf{x}'}\mathbf{x}' = \frac{9}{26}\mathbf{y} + \frac{19/2}{15/2}\mathbf{x}' = \frac{9}{26}\begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix} + \frac{19}{15}\begin{bmatrix} -1 \\ 1/2 \\ 5/2 \end{bmatrix}$$

$$= \frac{1}{195}\begin{bmatrix} 23 \\ 326 \\ 685 \end{bmatrix},$$

which is not equal to $\mathbf{b}_2$, so $\mathbf{b}_2 \notin C(A')$.

We have shown that $A'\mathbf{x} = \mathbf{b}_1$ has a solution (and the *method* even gave us a solution) whereas $A'\mathbf{x} = \mathbf{b}_2$ has no solution. Since "most" 3-vectors $\mathbf{b}$ do *not* lie in the plane $C(A')$, $A'\mathbf{x} = \mathbf{b}$ "usually" has no solution. In Chapter 22 we will present a systematic method to find all solutions to linear systems (when a solution exists). ■

Remark 21.2.8. Example 21.2.7 gives a good illustration of the fact that if $C(A)$ is a *proper* subspace (i.e., not the entirety) of $\mathbf{R}^m$ then $A\mathbf{x} = \mathbf{b}$ rarely has a solution. There is a solution precisely when the m-vector $\mathbf{b}$ lies in the proper subspace $C(A) \subset \mathbf{R}^m$ (a "rare" event when $C(A) \neq \mathbf{R}^m$).

For an $m \times n$ matrix A, the linear system $A\mathbf{x} = \mathbf{b}$ has a solution precisely when $\mathbf{Proj}_{C(A)}(\mathbf{b}) = \mathbf{b}$. The projection $\mathbf{Proj}_{C(A)}(\mathbf{b})$ can be computed using an orthogonal basis of $C(A)$ found via applying Gram–Schmidt to the columns of A. Unraveling the computation yields a solution to $A\mathbf{x} = \mathbf{b}$ when one exists, as illustrated in (21.2.7).

The somewhat algebraic-looking concept of the column space $C(A)$ of an $m \times n$ matrix A can be thought about in a more visual manner in terms of the corresponding linear transformation $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ defined by $\mathbf{f}(\mathbf{x}) = A\mathbf{x} \in \mathbf{R}^m$ for every $\mathbf{x} \in \mathbf{R}^n$, as follows.

The columns of A are precisely the outputs when we apply $\mathbf{f}$ to the standard basis vectors of $\mathbf{R}^n$: the first column $\mathbf{a}_1$ equals $\mathbf{f}(\mathbf{e}_1)$, the second column $\mathbf{a}_2$ equals $\mathbf{f}(\mathbf{e}_2)$, and so on (see Theorem 13.4.5). Now consider a vector $x_1\mathbf{a}_1 + \ldots + x_n\mathbf{a}_n$ in the span of these n columns of A. We can rewrite this as $x_1\mathbf{f}(\mathbf{e}_1) + \ldots + x_n\mathbf{f}(\mathbf{e}_n)$, and the apply the rules of linearity to get that

$$x_1\mathbf{a}_1 + \ldots + x_n\mathbf{a}_n = x_1\mathbf{f}(\mathbf{e}_1) + \ldots + x_n\mathbf{f}(\mathbf{e}_n) = \mathbf{f}(x_1\mathbf{e}_1 + \ldots + x_n\mathbf{e}_n).$$

However,

$$x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + \cdots + x_n\mathbf{e}_n = \begin{bmatrix} x_1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ x_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \ldots \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix},$$

which is just the vector which we have been calling $\mathbf{x}$. So all of this just expresses anew the fact that as we let $\mathbf{x}$ vary over all possible vectors in $\mathbf{R}^n$, then $A\mathbf{x}$ varies over all linear combinations of the columns of A. Thus, $C(A)$ consists of all the outputs of the linear transformation $\mathbf{f}$. (This recovers in another way the observation just after Definition 21.2.1 that $C(A)$ consists of exactly those $\mathbf{b} \in \mathbf{R}^m$ for which the vector equation $A\mathbf{x} = \mathbf{b}$ has a solution $\mathbf{x} \in \mathbf{R}^n$.) Since linear transformations can perhaps be visualized slightly better than matrices (e.g., rotations or shears in $\mathbf{R}^2$), this gives a more visual way to think about the outputs of $\mathbf{f}$. There is even some terminology for this:

Definition 21.2.9. The *image* of a linear transformation $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ is the collection of vectors $\mathbf{b} \in \mathbf{R}^m$ obtained as output of $\mathbf{f}$. In other words, the image of $\mathbf{f}$ consists of all vectors $\mathbf{b}$ of the form $\mathbf{b} = \mathbf{f}(\mathbf{x})$ for some $\mathbf{x} \in \mathbf{R}^n$. (This coincides with $C(A)$ for the $m \times n$ matrix A associated with $\mathbf{f}$.)

Although the "image" of a linear transformation is the same as the column space of the corresponding matrix, the idea it suggests in your mind may be more vivid. The image is always a linear subspace, since it is a column space by another name (or more specifically, it is the span of the collection of vectors $\mathbf{f}(\mathbf{e}_1), \dots, \mathbf{f}(\mathbf{e}_n)$). We give two examples: one geometric (to help with visualization) and one algebraic.

Example 21.2.10. If V is a linear subspace of $\mathbf{R}^n$ and A is the $n \times n$ matrix corresponding to the linear transformation $\mathbf{Proj}_V : \mathbf{R}^n \to \mathbf{R}^n$ then the image of $\mathbf{Proj}_V$, or equivalently the column space $C(A)$, is equal to V. Indeed, by definition $\mathbf{Proj}_V(\mathbf{x}) \in V$ for every $\mathbf{x} \in \mathbf{R}^n$, so the image is contained in V, and every $\mathbf{v} \in V$ is obtained in this way since $\mathbf{v} = \mathbf{Proj}_V(\mathbf{v})$ (indeed, the closest point in V to any $\mathbf{v} \in V$ is certainly $\mathbf{v}$). This is always a good example to have in mind when trying to visualize column spaces. ■

Example 21.2.11. Let us determine the image of the function $\mathbf{f} : \mathbf{R}^2 \to \mathbf{R}^3$ given by

$$\mathbf{f}\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 1 & -6 \\ 2 & 0 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = x \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + y \begin{bmatrix} -6 \\ 0 \\ 4 \end{bmatrix}.$$

As a first step, we describe the right side in words. Starting at the origin, walk with velocity $\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ for a time x, and then turn and walk with velocity $\begin{bmatrix} -6 \\ 0 \\ 4 \end{bmatrix}$ for time y. In Chapter 3, we saw in many examples that the vectors obtained this way constitute exactly a *plane passing through the origin*. So the image is some plane. But which plane is it?

We learned in Chapter 3 a few ways to describe a plane. We use one of these and find an equation for this plane by first finding a (nonzero) normal vector $\mathbf{n}$ to it. Recall that this is simply a vector $\mathbf{n}$ for which

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \cdot \mathbf{n} = 0, \quad \begin{bmatrix} -6 \\ 0 \\ 4 \end{bmatrix} \cdot \mathbf{n} = 0.$$

Writing $\mathbf{n} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ for some as yet unknown numbers a, b, c, the orthogonality conditions amount to the linear system,

$$a + 2b + 3c = 0, \quad -6a + 4c = 0$$

and we seek a solution $(a, b, c) \neq (0, 0, 0)$. Since the right side is zero for each equation, if $\mathbf{n}$ is one solution then $t\mathbf{n}$ is also a solution for any scalar t. Thus, hoping that there is a solution with *nonzero* third entry, we could take t to be the reciprocal of that entry and hence focus our search on a solution with $c = 1$. The resulting conditions on a and b become "2 equations in 2 unknowns" as in your prior study of algebra. In this case, we arrive at

$$a + 2b + 3 = 0, \quad -6a + 4 = 0.$$

The second of these gives $a = 4/6 = 2/3$, and inserting this into the first then becomes $(2/3)+2b+3 = 0$, or equivalently $b = -11/6$.

In other words, one such $\mathbf{n}$ is $\begin{bmatrix} 2/3 \\ -11/6 \\ 1 \end{bmatrix}$. Multiplying through by 6 to clear the denominators gives the slightly cleaner choice of normal vector

$$\mathbf{n} = \begin{bmatrix} 4 \\ -11 \\ 6 \end{bmatrix}.$$

The equation of the plane is then

$$\mathbf{n} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = 0,$$

which is the same as $4x - 11y + 6z = 0$.

To summarize: the image of $\mathbf{f}$ is the plane $4x - 11y + 6z = 0$. ■

21.3. Null space. We now turn to question (ii) in the list of main questions to be addressed:

(ii) If $A\mathbf{x} = \mathbf{b}$ has a solution, is there more than one? How do we describe the set of all solutions?

To get started, consider the linear system $A\mathbf{x} = \mathbf{b}$, where A is an $m \times n$ matrix and $\mathbf{b} \in \mathbf{R}^m$. Suppose we have, by some method or another, found two different solutions, $\mathbf{x}_0, \mathbf{x}_1 \in \mathbf{R}^n$. Is it possible that these are the only two solutions, or must there be more? How might we have predicted that there would be more than one solution?

To answer these two questions, observe that if we subtract the equation $A\mathbf{x}_0 = \mathbf{b}$ from the equation $A\mathbf{x}_1 = \mathbf{b}$ then we get

$$A(\mathbf{x}_1 - \mathbf{x}_0) = A\mathbf{x}_1 - A\mathbf{x}_0 = \mathbf{b} - \mathbf{b} = \mathbf{0}.$$

In other words, the vector $\mathbf{d} = \mathbf{x}_1 - \mathbf{x}_0$ solves $A\mathbf{d} = \mathbf{0}$. The fact that $\mathbf{x}_0 \neq \mathbf{x}_1$ means that $\mathbf{d} \neq \mathbf{0}$, whereas the right side of both equations was the same vector $\mathbf{b}$, so when we subtracted we got $\mathbf{0}$ on the right side.

What we have shown is that if there are two *different* solutions to $A\mathbf{x} = \mathbf{b}$, then there is necessarily a *nonzero* solution $\mathbf{d} \in \mathbf{R}^n$ to $A\mathbf{d} = \mathbf{0}$.

Example 21.3.1. In Example 21.2.7, with

$$A' = \begin{bmatrix} 2 & 1 & 4 \\ 4 & 2 & 3 \\ 6 & 3 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -3 \\ 4 \\ 13 \end{bmatrix},$$

we gave you three different solutions to $A'\mathbf{x} = \mathbf{b}$:

$$\mathbf{x}_0 = \begin{bmatrix} 0 \\ 5 \\ -2 \end{bmatrix}, \quad \mathbf{x}_1 = \begin{bmatrix} 1 \\ 3 \\ -2 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} -2 \\ 9 \\ -2 \end{bmatrix}.$$

By the calculation above, each of the nonzero difference vectors

$$\mathbf{x}_1 - \mathbf{x}_0 = \begin{bmatrix} 1 \\ -2 \\ 0 \end{bmatrix}, \quad \mathbf{x}_2 - \mathbf{x}_0 = \begin{bmatrix} -2 \\ 4 \\ 0 \end{bmatrix}, \quad \mathbf{x}_2 - \mathbf{x}_1 = \begin{bmatrix} -3 \\ 6 \\ 0 \end{bmatrix}$$

satisfies $A'\mathbf{x} = \mathbf{0}$ (please also directly check this!). ■

All of this motivates the fact that understanding whether there is *more than one* solution to $A\mathbf{x} = \mathbf{b}$ is very closely related to the question of whether there is any *nonzero* solution to

$$A\mathbf{x} = \mathbf{0}.$$

This latter equation is called the associated *homogeneous system* ("homogeneous" is a fancy word that refers to the fact that scalar multiplication takes solutions to solutions: if $A\mathbf{x} = \mathbf{0}$ and t is any scalar then $A(t\mathbf{x}) = t(A\mathbf{x}) = t\,\mathbf{0} = \mathbf{0}$). Any homogeneous system always has one "obvious" solution, namely $\mathbf{x} = \mathbf{0}$, since $A\mathbf{0} = \mathbf{0}$ (note here that the $\mathbf{0}$ on the left is the zero vector in $\mathbf{R}^n$ and the $\mathbf{0}$ on the right is the zero vector in $\mathbf{R}^m$). So what we learned above can be restated as saying that if the original linear system $A\mathbf{x} = \mathbf{b}$ has two *different* solutions then the associated homogenous system must have a *nonzero* solution.

We introduce some terminology for this:

Definition 21.3.2. The *null space* of A, denoted $N(A)$, is the set of all solutions in $\mathbf{R}^n$ to the homogeneous system $A\mathbf{x} = \mathbf{0}$.

Note that, in contrast to the column space, the null space $N(A)$ lies in $\mathbf{R}^n$, not $\mathbf{R}^m$.

A common question that arises is: how does one visualize null spaces? If we think about the associated linear transformation $T_A : \mathbf{R}^n \to \mathbf{R}^m$ defined by $T_A(\mathbf{x}) = A\mathbf{x}$ then $N(A)$ consists of everything that T_A sends to $\mathbf{0}$. So if T_A is an invertible transformation (such as a rotation or a shear) then $N(A) = \{\mathbf{0}\}$. An important case for which null spaces are much richer and can be visualized is projections:

Example 21.3.3. Consider the $n \times n$ matrix A corresponding to the linear transformation $\mathbf{Proj}_V : \mathbf{R}^n \to \mathbf{R}^n$, where V is a linear subspace of $\mathbf{R}^n$. We claim that $N(A)$ is the orthogonal complement $V^\perp$ (this is the collection of vectors $\mathbf{x} \in \mathbf{R}^n$ that are orthogonal to every vector in V, as in Theorem 19.2.5). Since $A\mathbf{x} = \mathbf{Proj}_V(\mathbf{x})$, this claim can be restated by saying that $\mathbf{Proj}_V(\mathbf{x}) = \mathbf{0}$ precisely when $\mathbf{x} \in V^\perp$. By Theorem 6.2.4, one of the characterizations of $\mathbf{Proj}_V(\mathbf{x})$ is that it is the unique vector in V whose difference from $\mathbf{x}$ is orthogonal to everything in V. But $\mathbf{0} \in V$, so it follows that $\mathbf{Proj}_V(\mathbf{x}) = \mathbf{0}$ precisely when $\mathbf{x} - \mathbf{0}$ is orthogonal to every vector in V, which is to say $\mathbf{x} \in V^\perp$. ■

Here is a more algebraic example, to be built upon in our subsequent discussion about null spaces.

Example 21.3.4. For the matrix A in Example 21.2.2, you can check directly that $N(A)$ contains the vector $\mathbf{v}_0 = \begin{bmatrix} 1/2 \\ -7 \\ 2 \end{bmatrix}$, as well as any scalar multiple of this vector. Likewise, for A' as in Example 21.2.3, check that $N(A')$ contains all scalar multiples of the vector $\mathbf{v}_0' = \begin{bmatrix} 1 \\ -2 \\ 0 \end{bmatrix}$. In fact, these are the complete descriptions of the null spaces in these two cases (i.e., if $\mathbf{x}$ lies in $N(A)$, then it must be a scalar multiple of $\mathbf{v}_0$, and similarly for A' using $\mathbf{v}_0'$), though that shouldn't be apparent to you yet. ■

We have produced these nonzero vectors in the preceding null spaces $N(A)$ and $N(A')$ out of thin air, and also asserted that these give a "complete description" of the null space in these two cases. You may wonder how we found these in the first place, and what it means in general to give a "complete description" of the null space (hopefully something better than having to explicitly list every single vector in it).

In Chapter 22 we will provide a systematic method to find the vectors in the null space of any given matrix A. We now say a bit about how to give a "complete description" of a null space. For any $m \times n$ matrix A, the zero vector $\mathbf{0} \in \mathbf{R}^n$ *always* lies in $N(A)$. Furthermore, $N(A)$ satisfies the same properties as those listed for linear subspaces in Proposition 4.1.11:

Proposition 21.3.5. For any $m \times n$ matrix A, the null space $N(A) \subset \mathbf{R}^n$ contains $\mathbf{0}$. Also, if $\mathbf{x}_1, \dots, \mathbf{x}_k \in N(A)$ then any linear combination $c_1\mathbf{x}_1 + \cdots + c_k\mathbf{x}_k$ also belongs to $N(A)$.

The argument for this is just a direct calculation, much as we did for linear subspaces. Indeed, if $T_A : \mathbf{R}^n \to \mathbf{R}^m$ is the linear transformation corresponding to A (i.e., $T_A(\mathbf{v})$ is the matrix-vector product $A\mathbf{v} \in \mathbf{R}^m$ for each $\mathbf{v} \in \mathbf{R}^n$) and $\mathbf{x}_1, \dots, \mathbf{x}_k \in N(A)$ then for any scalars $c_1, \dots, c_k$,

$$\begin{aligned} A \sum_{j=1}^{k} c_j\mathbf{x}_j = T_A\left(\sum_{j=1}^{k} c_j\mathbf{x}_j\right) = \sum_{j=1}^{k} T_A(c_j\mathbf{x}_j) = \sum_{j=1}^{k} c_j\, T_A(\mathbf{x}_j) &= \sum_{j=1}^{k} c_j\, A\mathbf{x}_j \\ &= \sum_{j-1}^{k} (c_j\, \mathbf{0}) \\ &= \mathbf{0}, \end{aligned}$$

so $\sum_{j-1}^{k} c_j\mathbf{x}_j \in N(A)$. It is no coincidence that Propositions 4.1.11 and 21.3.5 look so similar. In fact:

Proposition 21.3.6. The null space $N(A)$ is *always a linear subspace* of $\mathbf{R}^n$.

To be more specific, since we *defined* "linear subspace" to mean the span of a finite collection of vectors, we are saying that $N(A)$ is always the span of some finite collection of vectors. In other words, there always exist finitely many solutions to $A\mathbf{x} = \mathbf{0}$ whose span gives all solutions.

There is a more general fact that explains why $N(A)$ is always a linear subspace of $\mathbf{R}^n$:

Theorem 21.3.7. Suppose that V is a subset of $\mathbf{R}^n$ with the following properties: $\mathbf{0} \in V$, and for all $\mathbf{x}_1, \dots, \mathbf{x}_k \in V$ and all scalars $c_1, \dots, c_k$ the n-vector $c_1\mathbf{x}_1 + \ldots + c_k\mathbf{x}_k$ always belongs to V. Then V must be the span of a finite collection of vectors, so (by definition) V is a *linear subspace* of $\mathbf{R}^n$.

Remark 21.3.8. A shorthand way of stating Theorem 21.3.7 is that if V is a subset of $\mathbf{R}^n$ containing $\mathbf{0}$ that is "closed" under scalar multiplication and vector addition (meaning that if $\mathbf{x}, \mathbf{y} \in V$ and c is any scalar then $c\mathbf{x} \in V$ and $\mathbf{x} + \mathbf{y} \in V$) then V is a linear subspace.

The characterization of linear subspaces in Theorem 21.3.7 is **extremely useful** in applications of linear algebra. It is discussed further in Section B.2, where its utility is illustrated for null spaces (as we have discussed here) and orthogonal complements, neither of which are obviously linear subspaces (according to our definition of "linear subspace"). In Section B.2 we also prove Theorem 21.3.7 (see Theorem B.2.6) for those who are interested. This answers what it means to give a "complete description" of $N(A)$: we produce a spanning set (even better, we might produce a basis for $N(A)$, or best yet an orthogonal basis by applying Gram–Schmidt to a spanning set).

Remark 21.3.9. In most textbooks on linear algebra, Theorem 21.3.7 is often taken as the *primary definition* of "linear subspace". This has advantages and disadvantages. Since the most practical way we have to describe subspaces is as spans of finite collections of vectors, we have chosen our more concrete definition to emphasize the fact that subspaces can be described in terms of a finite amount of data. However, once we know Theorem 21.3.7, we can reap the benefit of both perspectives (i.e., the span definition and Theorem 21.3.7); in some situations one perspective is easier to verify and in other situations the other is.

Now let us return to the main task of this section: describing all solutions to $A\mathbf{x} = \mathbf{b}$ with any $\mathbf{b}$ for which some solution exists. We saw how two different solutions of the same linear system yield a nonzero element $\mathbf{d} \in N(A)$. The reverse statement is true as well: if $A\mathbf{x}_0 = \mathbf{b}$ and if $\mathbf{d}$ is some nonzero element of $N(A)$ then $\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{d}$ is *another* solution to $A\mathbf{x} = \mathbf{b}$. Indeed,

$$A\mathbf{x}_1 = A(\mathbf{x}_0 + \mathbf{d}) = A\mathbf{x}_0 + A\mathbf{d} = \mathbf{b} + \mathbf{0} = \mathbf{b}.$$

So we have answered the main question: if there is more than one solution to $A\mathbf{x} = \mathbf{b}$ then $N(A)$ *must* be a nonzero subspace (by the subtraction technique) and in reverse if $N(A)$ contains a nonzero vector and $A\mathbf{x} = \mathbf{b}$ has at least one solution $\mathbf{x}_0$ (i.e., $\mathbf{b} \in C(A)$) then there are different solutions corresponding to every element of $N(A)$ (namely, $\mathbf{x}_0 + \mathbf{d}$ for every $\mathbf{d} \in N(A)$). Note that $N(A)$ contains infinitely many elements when it contains a nonzero vector (indeed, if $\mathbf{d} \in N(A)$ is a nonzero vector then any scalar multiple of $\mathbf{d}$ also lies in $N(A)$ and there are infinitely many of these). To summarize:

Proposition 21.3.10. For any $m \times n$ matrix A and $\mathbf{b} \in \mathbf{R}^m$ for which the vector equation $A\mathbf{x} = \mathbf{b}$ has *some* solution $\mathbf{x}_0 \in \mathbf{R}^n$, the solutions to $A\mathbf{x} = \mathbf{b}$ are precisely the vectors of the form $\mathbf{x}_0 + \mathbf{d}$ for $\mathbf{d} \in N(A)$. There are infinitely many solutions whenever $N(A)$ contains a nonzero vector.

Let's illustrate Proposition 21.3.10 algebraically, and then use it to give a helpful way to visualize what the collection of solutions to $A\mathbf{x} = \mathbf{b}$ looks like (when there is some solution!).

Example 21.3.11. Consider the 4×4 matrix

$$A = \begin{bmatrix} 2 & 1 & 3 & 3 \\ -1 & 1 & -3 & 0 \\ 5 & -2 & 12 & 3 \\ 3 & 7 & -1 & 10 \end{bmatrix}.$$

It turns out (using methods in Chapter 22) that the subspace $N(A) \subset \mathbf{R}^4$ is a plane, given explicitly as

$$N(A) = \mathrm{span}(\mathbf{v}, \mathbf{v}') = \{c\mathbf{v} + c'\mathbf{v}' : c, c' \in \mathbf{R}\} \text{ for } \mathbf{v} = \begin{bmatrix} -2 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{v}' = \begin{bmatrix} -1 \\ -1 \\ 0 \\ 1 \end{bmatrix}.$$

Hence, for any $\mathbf{b} \in \mathbf{R}^4$ for which there exists a solution $\mathbf{x}_0$ to $A\mathbf{x} = \mathbf{b}$, the collection of all solutions is the set of all vectors of the form $\mathbf{x}_0 + c\mathbf{v} + c'\mathbf{v}'$ for $c, c' \in \mathbf{R}$.

Let us see what this tells us for a specific $\mathbf{b}$, say $\mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ -2 \\ 7 \end{bmatrix}$. It turns out that this vector lies in $C(A)$, so $A\mathbf{x} = \mathbf{b}$ does have at least one solution. In Chapter 22 you will learn a systematic way to find solutions, but $\mathbf{x}_0 = \begin{bmatrix} -2 \\ 2 \\ 1 \\ 0 \end{bmatrix}$ is a solution. Hence, the collection of *all* solutions to $A\mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ -2 \\ 7 \end{bmatrix}$ is the set of all vectors of the form

$$\mathbf{x}_0 + c\mathbf{v} + c'\mathbf{v}' = \begin{bmatrix} -2 \\ 2 \\ 1 \\ 0 \end{bmatrix} + c\begin{bmatrix} -2 \\ 1 \\ 1 \\ 0 \end{bmatrix} + c'\begin{bmatrix} -1 \\ -1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -2-2c-c' \\ 2+c-c' \\ 1+c \\ c' \end{bmatrix} \quad \text{for } c, c' \in \mathbf{R}.$$

■

Let's now use the relationship between $N(A)$ and the set of all solutions to $A\mathbf{x} = \mathbf{b}$ to visualize the collection of such solutions. As we have said in several different ways already, if A is an $m \times n$ matrix and $\mathbf{b} \in \mathbf{R}^m$ is an m-vector for which $A\mathbf{x} = \mathbf{b}$ has *some* solution $\mathbf{x}_0$ then the collection of all solutions is described as the vectors of the form $\mathbf{x}_0 + \mathbf{v}$ with $\mathbf{v} \in N(A)$. We can visualize this easily as follows when $n = 3$ and $N(A)$ is either a line or a plane; i.e., $\dim N(A) = 1$ or 2. In either of these cases, as shown in Figure 21.3.1, the solutions to $A\mathbf{x} = \mathbf{b}$ constitute either a line or a plane in $\mathbf{R}^3$ *typically not going through* $\mathbf{0}$ (except when $\mathbf{b} = \mathbf{0}$).

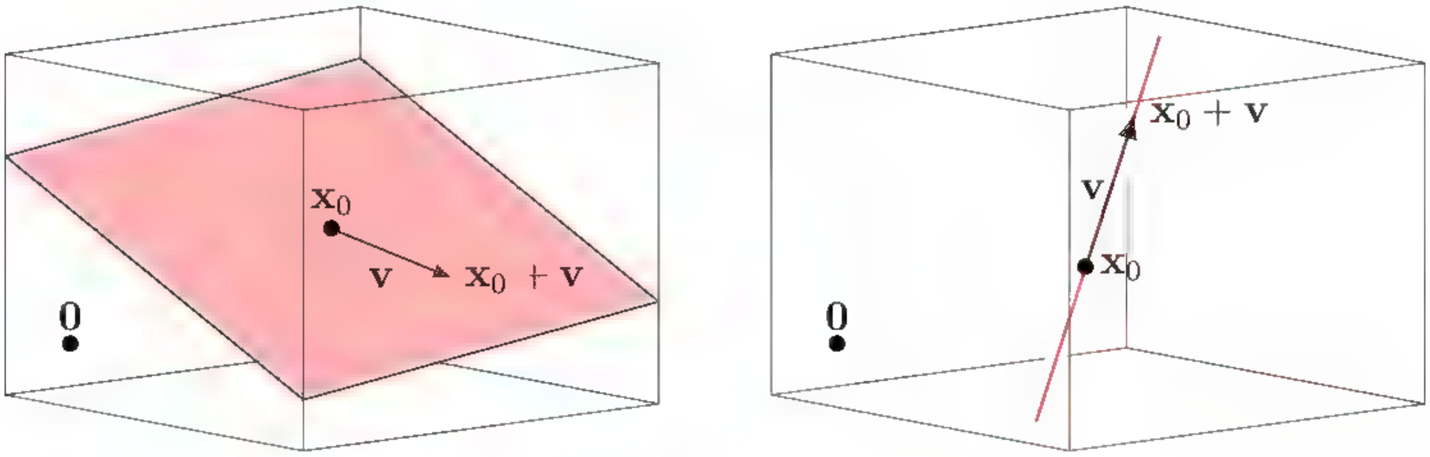


FIGURE 21.3.1. When $n = 3$, the collection of points $\mathbf{x}_0 + \mathbf{v}$ for all $\mathbf{v} \in N(A)$ is a plane through $\mathbf{x}_0$ (left) when $\dim N(A) = 2$ and is a line through $\mathbf{x}_0$ (right) when $\dim N(A) = 1$.

We repeat a special case of our conclusions for emphasis:

> If $A\mathbf{x} = \mathbf{b}$ has two different solutions (so $N(A)$ is nonzero, by Proposition 21.3.10) then it has infinitely many solutions: if $\mathbf{x}_0$ is a solution and $\mathbf{v} \in N(A)$ is nonzero then the entire line $\{\mathbf{x}_0 + t\mathbf{v} : t \in \mathbf{R}\}$ through $\mathbf{x}_0$ consists of solutions.

Example 21.3.12. For the 4×4 matrix A in Example 21.3.11, we exhibited a couple of nonzero vectors in $N(A)$, namely $\mathbf{v} = \begin{bmatrix} -2 \\ 1 \\ 1 \\ 0 \end{bmatrix}$ and $\mathbf{v}' = \begin{bmatrix} -1 \\ -1 \\ 0 \\ 1 \end{bmatrix}$. Hence, whenever $A\mathbf{x} = \mathbf{b}$ has some solution $\mathbf{x}_0$, we can get infinitely many more via vectors of the form $\mathbf{x}_0 + c\mathbf{v}$ for $c \in \mathbf{R}$, or $\mathbf{x}_0 + c\mathbf{v}'$ for $c \in \mathbf{R}$. (The vector $\mathbf{v} + \mathbf{v}'$ also lies in $N(A)$ and is nonzero by inspection, so the vectors $\mathbf{x}_0 + c(\mathbf{v} + \mathbf{v}')$ for $c \in \mathbf{R}$ are another supply of infinitely many solutions to $A\mathbf{x} = \mathbf{b}$.) ■

We conclude this section with a worked example of a linear system where we must first determine $C(A)$ and after that $N(A)$, and then use these two subspaces to understand the solvability of the equation $A\mathbf{x} = \mathbf{b}$ for some given $\mathbf{b}$.

Example 21.3.13. Consider the system of linear equations

$$3x + 5y - z = 7, \quad -x + y + 3z = 3, \quad 2x + 3y - z = 4,$$

which has the equivalent matrix formulation

$$A \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 & 5 & -1 \\ -1 & 1 & 3 \\ 2 & 3 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 7 \\ 3 \\ 4 \end{bmatrix}.$$

Let us determine whether this vector equation has a solution, and if so then find all of them. We approach this in 4 steps, using the general ideas discussed above.

Step 1: compute $C(A)$. The *column space* of A is the span of the three vectors

$$\mathbf{v} = \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix}, \quad \mathbf{v}' = \begin{bmatrix} 5 \\ 1 \\ 3 \end{bmatrix}, \quad \mathbf{v}'' = \begin{bmatrix} -1 \\ 3 \\ -1 \end{bmatrix}.$$

Is the span all of $\mathbf{R}^3$, or equivalently are these linearly independent? It happens to be the case that $\mathbf{v}'' = \mathbf{v}' - 2\mathbf{v}$ (check); don't worry about how one might have discovered this (linear dependence amounts to some linear combination $x\mathbf{v} + y\mathbf{v}' + z\mathbf{v}'' = \mathbf{0}$ with x, y, z not all 0, which is just another way of saying that $N(A)$ is nonzero; the task of actually computing $N(A)$ systematically is addressed in Chapter 22).

Hence, any vector which can be written as a linear combination of these three vectors can be written more simply as a linear combination of two of them:

$$x\mathbf{v} + y\mathbf{v}' + z\mathbf{v}'' = x\mathbf{v} + y\mathbf{v}' + z(\mathbf{v}' - 2\mathbf{v}) = (x - 2z)\mathbf{v} + (y + z)\mathbf{v}' = (x - 2z)\begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix} + (y + z)\begin{bmatrix} 5 \\ 1 \\ 3 \end{bmatrix}.$$

We can make $x - 2z$ and $y + z$ take on whatever two values we wish (e.g., set $z = 0$ and assign whatever we want to x and y), so

$$C(A) = \mathrm{span}(\mathbf{v}, \mathbf{v}') = \{c\mathbf{v} + c'\mathbf{v}' : c, c' \in \mathbf{R}\} = \left\{ \begin{bmatrix} 3c + 5c' \\ -c + c' \\ 2c + 3c' \end{bmatrix} : c, c' \in \mathbf{R} \right\}.$$

Step 2: use $C(A)$ to check if some solution $\mathbf{x}_0$ exists. Now let us match this with our particular right side $\mathbf{b} = \begin{bmatrix} 7 \\ 3 \\ 4 \end{bmatrix}$. A solution exists to $A\mathbf{x} = \mathbf{b}$ precisely when $\mathbf{b} \in C(A) = \mathrm{span}(\mathbf{v}, \mathbf{v}')$. There are two ways to think about this: either explicitly that we can find $c, c' \in \mathbf{R}$ for which $\mathbf{b} = c\mathbf{v} + c'\mathbf{v}'$, which is to say

$$\begin{bmatrix} 7 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 3c + 5c' \\ -c + c' \\ 2c + 3c' \end{bmatrix},$$

or alternatively that $\mathbf{Proj}_{C(A)}(\mathbf{b}) = \mathbf{b}$. The latter viewpoint is the systematic way to approach the task, as we have discussed earlier, but it has the disadvantage (with our present state of knowledge) that it doesn't produce an explicit $\mathbf{x}_0$.

So in the present case where the dimensions are small we'll use the former viewpoint that amounts to 3 equations in 2 unknowns:

$$3c + 5c' = 7, \quad -c + c' = 3, \quad 2c + 3c' = 4.$$

We can solve the first two of these simultaneously via the procedure from your prior study of algebra and then check if it also satisfies the third condition too: the first two have as the simultaneous solution exactly $(c, c') = (-1, 2)$, and this indeed satisfies the third condition. So the vector equation $A\mathbf{x} = \mathbf{b}$ does have a solution $\mathbf{x}_0 = \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix}$ (corresponding to the expression $-\mathbf{v} + 2\mathbf{v}' + 0\mathbf{v}''$ just obtained for $\mathbf{b}$).

Step 3: compute $N(A)$: Next we turn to the other part of the problem: describing all solutions when at least one exists. Again following our general ideas, we first compute the null space of A; i.e., the set of all solutions to

$$A\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{21.3.1}$$

The systematic way to find $N(A)$ will be discussed in Chapter 22. It turns out that $N(A) \subset \mathbf{R}^3$ is a line: the span of $\begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$ (this is related to the fact that the third column $\mathbf{v}''$ is equal to $\mathbf{v}' - 2\mathbf{v}$, but don't worry about this).

Step 4: combine Steps 2 and 3 to write down the general solution. Finally, the solutions to the original linear system are precisely the vectors of the form

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{d}, \quad \text{where } \mathbf{d} \in N(A).$$

Using the numbers we have found, the vectors of the form

$$\mathbf{x} = \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix} + t\begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1+2t \\ 2-t \\ t \end{bmatrix}$$

for $t \in \mathbf{R}$ are precisely the solutions. Breaking this down into components, any solution (x, y, z) is of the form $x = -1 + 2t$, $y = 2 - t$, $z = t$ where t is *the same* number in all three expressions.

■

We summarize the conclusions reached so far, using the language introduced above.

Theorem 21.3.14. Let A be an $m \times n$ matrix. The equation $A\mathbf{x} = \mathbf{b}$ has:

(i) no solution if $\mathbf{b}$ does not belong to the column space $C(A)$ of A,

(ii) exactly one solution if $\mathbf{b} \in C(A)$ and the null space $N(A)$ consists of only the zero vector,

(iii) infinitely many solutions if $\mathbf{b}$ belongs to $C(A)$ and the null space $N(A)$ is nonzero (in which case the solutions are precisely the vectors $\mathbf{x}_0 + \mathbf{v}$ for all $\mathbf{v} \in N(A)$, where $\mathbf{x}_0$ is any single solution).

To conclude our general discussion of null spaces, we would like to discuss a "conservation law" for dimension that provides a precise sense in which the "size" of $N(A)$ is tightly linked to the "size" of $C(A)$, where our sense of "size" will be taken to mean dimension. A numerical example conveys the key intuition quite well:

Example 21.3.15. Let A be an 11×7 matrix, so the associated linear transformation $L = T_A$ is a function $\mathbf{R}^7 \to \mathbf{R}^{11}$. The null space $N(A)$ is a linear subspace of $\mathbf{R}^7$ (it consists of the vectors annihilated by L; i.e., sent to $\mathbf{0}$), and the column space $C(A)$ is a linear subspace of $\mathbf{R}^{11}$ (it consists of the output of the transformation L; i.e., it is the image of L).

Certainly $\dim N(A) \leq 7$ and $\dim C(A) \leq \mathbf{R}^{11}$ since $N(A) \subset \mathbf{R}^7$ and $C(A) \subset \mathbf{R}^{11}$. But there is something much better which can be said:

$$\dim C(A) + \dim N(A) = 7.$$

In particular, $\dim C(A) \leq 7$ (which might seem surprising: $\dim C(A) = 9$ is impossible, for example) and if we happen to know that $\dim C(A) = 4$ (e.g., applying Gram–Schmidt to the columns of A to compute $\dim C(A)$) then we are claiming that automatically $\dim N(A) = 7 - 4 = 3$. Why is this true?

If we think in terms of the matrix A then this is rather baffling, but if we think visually in terms of $L : \mathbf{R}^7 \to \mathbf{R}^{11}$ then there is a very intuitive explanation in the spirit of a conservation law as follows. The transformation L is applied to inputs from $\mathbf{R}^7$ that have 7 "degrees of freedom" (intuitively, "degrees of freedom" is the number of "independent" directions of motion; it is made precise by the mathematical concept of dimension). By the meaning of $N(A)$ as the space of vectors annihilated by L, when we apply L to the input space $\mathbf{R}^7$ we lose $\dim N(A)$ degrees of freedom (more precisely, $L(\mathbf{x}) = L(\mathbf{x}+\mathbf{d})$ for any $\mathbf{d} \in N(A)$, so $L(\mathbf{x})$ only "remembers" $\mathbf{x}$ up to the ambiguity of $N(A)$). For example, if L annihilates exactly a plane (i.e., $\dim N(A) = 2$) then the output of L should have 2 fewer "degrees of freedom" than the space $\mathbf{R}^7$ of inputs.

The upshot is that the space of outputs of L should have as its number of "degrees of freedom" exactly $\dim N(A)$ less than the number of "degrees of freedom" of the space $\mathbf{R}^7$ of inputs. But the space of outputs is $C(A)$, so by interpreting "degrees of freedom" to really mean dimension this is saying $\dim C(A) = 7 - \dim N(A)$, or equivalently $\dim C(A) + \dim N(A) = 7$ as claimed. ■

There is nothing special about 7 (or 11) in the preceding example. By the same informal reasoning, if A is any $m \times n$ matrix then the transformation $L = T_A : \mathbf{R}^n \to \mathbf{R}^m$ applied to a space of inputs with n degrees of freedom annihilates $\dim N(A)$ degrees of freedom, so the space $C(A)$ of outputs of L should have its number of degrees of freedom equal to $\dim N(A)$ less than the number n that we start with for the input space. In other words, this suggests that $\dim C(A) = n - \dim N(A)$, or equivalently:

Theorem 21.3.16 (Rank–Nullity Theorem). For every $m \times n$ matrix A, $\dim C(A) + \dim N(A) = n$.

(The name is due to $\dim C(A)$ being called the *rank* of A and $\dim N(A)$ being called the *nullity* of A.)

This theorem is proved in Section 21.5 if you are interested. Hopefully the preceding visualization for this in terms of behavior of "degrees of freedom" under the effect of a linear transformation makes it very plausible. Here is a nice application:

Example 21.3.17. Let's revisit the linear system $A'\mathbf{x} = \mathbf{b}_1$ from Example 21.2.7, for which we found a solution $\mathbf{x}_0 = \begin{bmatrix} 0 \\ 5 \\ -2 \end{bmatrix}$. In that work, we determined $\dim C(A') = 2$, so the Rank–Nullity Theorem tells us that $\dim N(A') = 3 - 2 = 1$; i.e., $N(A')$ is a line. The collection of all solutions consists of vectors of the form $\mathbf{x}_0 + \mathbf{v}$ for $\mathbf{v} \in N(A')$, and if we find a nonzero $\mathbf{v}_0 \in N(A')$ then the line $N(A')$ consists of all scalar multiples $t\mathbf{v}_0$, so the solutions to $A'\mathbf{x} = \mathbf{b}_1$ consist of the vectors $\mathbf{x}_0 + t\mathbf{v}_0$. By inspection of A' we see that the 1st column is twice the 2nd column, so the vector $\mathbf{v}_0 = \begin{bmatrix} 1 \\ -2 \\ 0 \end{bmatrix}$ belongs to $N(A')$. (In Chapter 22 we will discuss systematic ways to describe null spaces, without requiring clever observations or "inspection".) Hence, the solutions to $A'\mathbf{x} = \mathbf{b}_1$ are the vectors of the form

$$\mathbf{x}_0 + t\mathbf{v}_0 = \begin{bmatrix} 0 \\ 5 \\ -2 \end{bmatrix} + t \begin{bmatrix} 1 \\ -2 \\ 0 \end{bmatrix} = \begin{bmatrix} t \\ 5 - 2t \\ -2 \end{bmatrix}$$

for $t \in \mathbf{R}$. (The values $t = 1$ and $t = -2$ recover the two additional solutions beyond $\mathbf{x}_0$ that were mentioned in Example 21.2.7.) ■

21.4. Overdetermined, underdetermined, and overfitting.

There is a "rule of thumb" about when we expect a linear system to have no solutions, a single solution, or infinitely many solutions. In very broad terms, we say that a system of m equations in n variables is *underdetermined* if there are fewer equations than unknowns (i.e., $m < n$), and *overdetermined* if there are more equations than unknowns (i.e., $m > n$). We expect (but are not guaranteed!) that underdetermined systems have infinitely many solutions and overdetermined systems have no solutions.

One way to think about this is that each additional linear equation in the system reduces the "degrees of freedom" (whatever that may mean) by 1. If we start out with no equations, then there are n degrees of freedom (no equations means no restrictions on $\mathbf{R}^n$); as we impose equations, we lower the degrees of freedom. If we impose fewer than n equations (when there are n variables) then we expect to have remaining degrees of freedom ($n - m$ degrees of freedom for m equations with $m < n$), whereas if we impose too many equations then we expect to have made too many requirements and so there should be no solution.

Example 21.4.1. A typical linear system of 3 equations in 5 unknowns should have infinitely many solutions (and $5 - 3 = 2$ "degrees of freedom", whatever that may mean), whereas a typical linear system of 5 equations in 3 unknowns should have no solutions ("too many equations on $\mathbf{R}^3$"). ■

But be careful, these are indeed only expectations. It is called a "rule of thumb" because counting the number of equations and number of unknowns is too rough to give definite conclusions for any specific linear system. When there are the *same* number of equations as unknowns, we expect that there is exactly one solution. However, as we have been discussing at length, this need not be the case!

Definition 21.4.2. A linear system $A\mathbf{x} = \mathbf{b}$ with m equations and n unknowns (i.e., "variables" $x_1, \ldots, x_n$) is called:

- *overdetermined* if there are more equations than unknowns, which is to say $m > n$;
- *underdetermined* if there are fewer equations than unknowns, which is to say $m < n$.

Remark 21.4.3. For underdeterminedness, think about a pair of different planes in $\mathbf{R}^3$ (not necessarily passing through the origin), corresponding to 2 equations in 3 unknowns (with $2 < 3$). Typically these will cross each other in exactly a line (the first picture in Figure 21.4.1), but in special circumstances the planes may be parallel and so not touch at all (the second picture in Figure 21.4.1). Informally, this says that 2 equations in 3 unknowns typically have lots of solutions (a line where the planes cross), but in special cases there is no solution (so the "rule of thumb" fails).

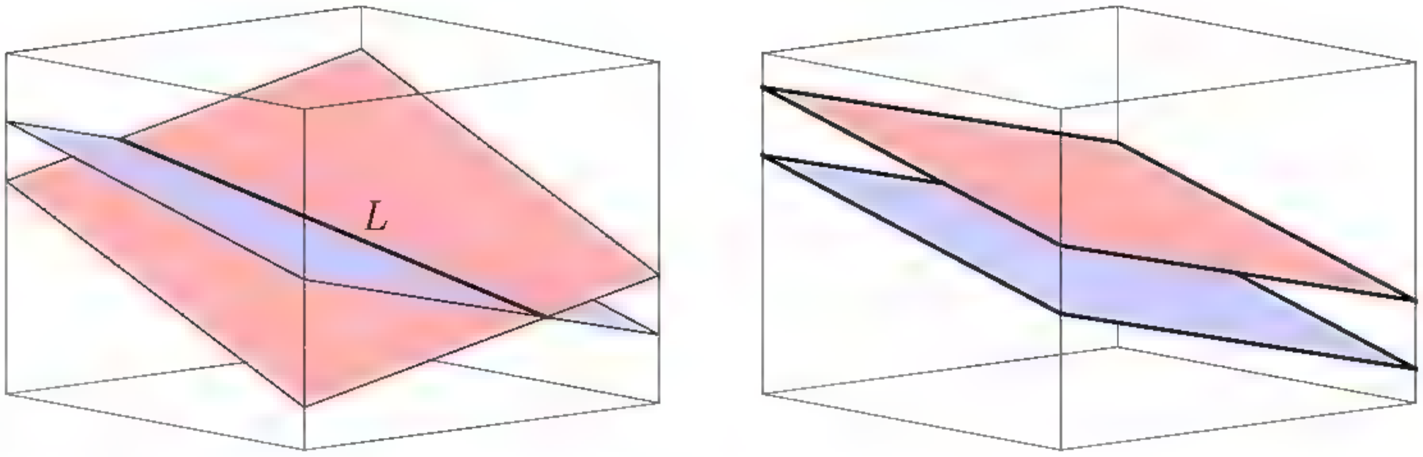


FIGURE 21.4.1. A pair of planes in $\mathbf{R}^3$ (not necessarily through the origin) is *underdetermined*: usually they meet along a line L (left), but in rare cases they don't touch (right).

For overdeterminedness, think about a pair of different lines in $\mathbf{R}^3$ (not necessarily passing through the origin), as in Figure 21.4.2. Each line in $\mathbf{R}^3$ corresponds to two simultaneous equations in 3 unknowns, so a point on both lines corresponds to a solution to a combined system of 4 equations in 3 unknowns. This combined system is overdetermined ($4 > 3$), and typically such lines don't touch at all, as in the first picture in Figure 21.4.2; this expresses the rule of thumb that overdetermined systems usually have no solution. But in special circumstances such a pair of distinct lines do meet at a point, as in the second picture in Figure 21.4.2, so in such cases there is a solution (and hence the "rule of thumb" on overdetermined systems can fail, though such circumstances are "special").

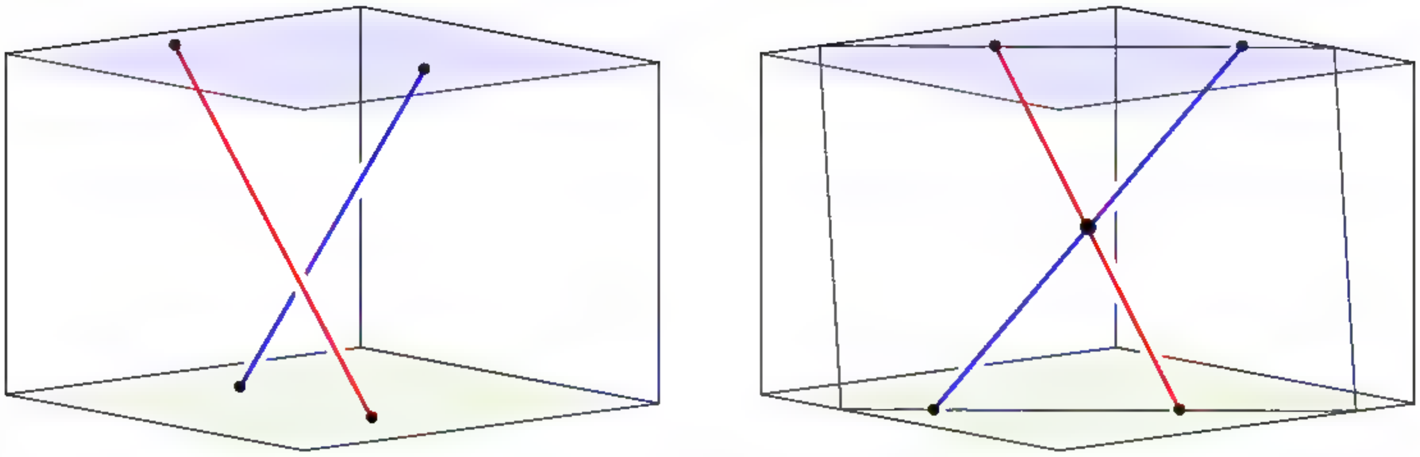


FIGURE 21.4.2. Two lines which don't meet (left), and two lines which do meet (right).

Remark 21.4.4. Our concept of dimension gives a way to provide precise reasons underlying the expectations in the overdetermined and underdetermined cases, as follows.

In the overdetermined case ($m > n$), we claim that for dimension reasons *necessarily* the column space $C(A) \subset \mathbf{R}^m$ is a proper subspace (i.e., it does not exhaust the entirety of $\mathbf{R}^m$, so the visualization is like a line or plane through the origin in $\mathbf{R}^3$). To see this, first observe that $C(A)$ is always spanned by the n columns of A, so $\dim C(A) \leq n$. Hence, for overdetermined cases we have $\dim C(A) \leq n < m$, so indeed $C(A)$ cannot be all of $\mathbf{R}^m$. To visualize such $C(A)$ inside $\mathbf{R}^m$, think about a line or plane through the origin in $\mathbf{R}^3$, outside of which are "most" of the vectors in $\mathbf{R}^3$; likewise in general "most" vectors $\mathbf{b} \in \mathbf{R}^m$ lie outside the lower-dimensional $C(A)$, and $A\mathbf{x} = \mathbf{b}$ has no solution in such cases. Informally, there are too many equations for the number of unknowns, so a solution $\mathbf{x}$ can (and "usually" does) fail to exist.

In the underdetermined case ($m < n$), we claim that for dimension reasons *necessarily* the null space $N(A)$ is nonzero. To see this, observe that the condition for a vector $\mathbf{x} \in \mathbf{R}^n$ to lie in $N(A)$ is the same as $\mathbf{x}$ being orthogonal to each of the m rows of A. In other words, if $V \subset \mathbf{R}^n$ is the span of the m rows then $N(A) = V^\perp$. But $\dim V \leq m$ since V is spanned by m vectors, so by the relationship in Theorem 19.2.5 between the dimensions of a linear subspace and its orthogonal complement in $\mathbf{R}^n$ we have

$$\dim N(A) = \dim V^\perp = n - \dim(V) \geq n - m > 0. \tag{21.4.1}$$

This shows that $N(A)$ contains nonzero vectors. By definition, then, the associated homogenous system $A\mathbf{x} = \mathbf{0}$ has nonzero solutions. Hence, if $A\mathbf{x} = \mathbf{b}$ has a solution $\mathbf{x}_0$ then it has infinitely many solutions (add to any single solution $\mathbf{x}_0$ the scalar multiples of a nonzero vector in $N(A)$).

Keep in mind however that even if $m < n$, **there might be no solutions** (i.e., there may be no $\mathbf{x}_0$ with which to get started)! This can happen if the equations are mutually inconsistent, as occurs for the system

$$x + y = 5, \quad 2x + 2y = 6$$

(a pair of parallel lines). Observe that the solutions of each of these two equations individually is a line in $\mathbf{R}^2$, and the inconsistency is that these lines are parallel and so do not intersect, which means that there is no *simultaneous* solution.

We need to know that there is *some* solution $\mathbf{x}_0$ to $A\mathbf{x} = \mathbf{b}$ before we can use that $N(A)$ contains many vectors in order to build infinitely many solutions (as $\mathbf{x}_0 + \mathbf{v}$ for $\mathbf{v} \in N(A)$). To summarize:

Remark 21.4.5. (i) In the overdetermined case, the linear system $A\mathbf{x} = \mathbf{b}$ often fails to have any solutions at all (namely, whenever $\mathbf{b}$ lies outside the linear subspace $C(A) \neq \mathbf{R}^m$). Informally, there are too many equations (but in special circumstances there can still be a solution). This occurs *a lot* in applications, and then we seek a "best approximate solution"; see Example 22.5.3.

(ii) In the underdetermined case, *if* $A\mathbf{x} = \mathbf{b}$ *has a solution* then it automatically has infinitely many solutions. (Informally, there are too few equations, so they do not pin down the values of all of the unknowns, if there is some solution at all.)

Example 21.4.6. Example 21.2.2 corresponds to 2 equations in 3 unknowns, so it is underdetermined. This is consistent with the fact that its null space is nonzero (an explicit nonzero vector in the null space for this case is $(1,-14,4)$), which enables us to build infinitely many solutions to $A\mathbf{x} = \mathbf{b}$ whenever there is at least one solution $\mathbf{x}_0$ (by adding to $\mathbf{x}_0$ any nonzero scalar multiple of a chosen nonzero $\mathbf{v}_0 \in N(A)$).

In contrast, consider the system $3x + 5y = 7$, $-x + y = 3$, $2x + 3y = 4$ of 3 equations in 2 unknowns. In matrix language, this is $A\mathbf{x} = \mathbf{b}$ for $A = \begin{bmatrix} 3 & 5 \\ -1 & 1 \\ 2 & 3 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} 7 \\ 3 \\ 4 \end{bmatrix}$. This is overdetermined, so for "typical" $\mathbf{b}' \in \mathbf{R}^3$ we expect that $A\mathbf{x} = \mathbf{b}'$ has no solution at all. For the specific $\mathbf{b}$ we have chosen, a solution *does* exist: $(x,y) = (-1,2)$. This is really quite special: the column space $C(A) \subset \mathbf{R}^3$ is the span of the two columns of A which (by inspection) are nonzero and not scalar multiples of each other, so $C(A)$ is a plane. Hence, for $\mathbf{b}'$ outside that plane (which is "typical" for vectors in $\mathbf{R}^3$), the vector equation $A\mathbf{x} = \mathbf{b}'$ has no solution. (Explicitly, this plane turns out to consist of those vectors $\mathbf{b}' \in \mathbf{R}^3$ satisfying $5b'_1 - b'_2 - 8b'_3 = 0$, a condition that is indeed satisfied by $\mathbf{b} = (7,3,4)$.) ■

Remark 21.4.7 (online resource). A nice dynamic visual overview of the concepts introduced in this chapter is given in this video at "Essence of Linear Algebra".

Example 21.4.8 (Overfitting). A significant consequence of this analysis is the danger of *overfitting*. This refers to the fact that if you have a mathematical model for some situation which is written as a linear system but the model has more unknowns than equations, then you will (usually) find many solutions but these are almost surely "physically meaningless". The error in designing the model is that you have incorporated too many variables for the number of equations.

Here is an example. To beat the stock market, let's predict the value of a specific stock on December 31 by imagining that it depends linearly on the highest temperatures at each day of the same year between January 1 and December 25 (giving us 6 days at the end of the year to use this prediction to buy or sell stocks in this company). This is an absurd idea: there should be no correlation at all between temperatures and stock prices. But suppose we stubbornly go ahead and try to *find* such a model fitting many years of data (and then make "predictions"!).

Ignoring leap years, mathematically what we are doing is seeking 359 ($= 365 - 6$) scalars $a_1, \dots, a_{359}$ for which

$$\text{stock price on Dec. 31} = a_1 t_1 + a_2 t_2 + \cdots + a_{359} t_{359}, \tag{21.4.2}$$

where t_j is the maximum temperature on the jth day of the year. Say we have temperature data for the past 50 years as well as records of the stock's price on December 31 of each of these same 50 years. We do not know values of the a_j's, but if we can find such values fitting the historic data then when we get to December 25 of this year, we'll use (21.4.2) to "predict" the stock's price several days later.

Fitting the historic data amounts to a system of 50 linear equations in 359 unknowns $a_1, \dots, a_{359}$. This is vastly underdetermined and so nearly always (i.e., except for the types of incompatibilities we mentioned above, which are rare) there will be infinitely many solutions. Upon choosing one of these solutions $(a_1, \dots, a_{359})$, the stock price "predicted" by (21.4.2) would work for the past 50 years (due to how we found the a_j's!) but it would be insane to rely on such a "prediction" for this year. ■

We can often create a mathematical model with some number of equations and unknowns, but if this model has too many unknowns for the number of equations then solutions fitting a lot of data are typically "forced to exist" for purely mathematical reasons that have no real meaning in terms of the model and so have no real significance for the intended applications of the model.

Overfitting can occur in many situations, not just for linear systems. There are instances across many fields where one is tempted to fit data using an overly complex model that contains too many unknowns relative to the number of (hopefully meaningful) constraints imposed. This is a huge problem in machine learning (e.g., see [**Ta**]) and a source of many investment scams [**St**]. Don't be fooled by "big data" hype!

Example 21.4.9. Here is another example of overfitting. It is easy to find coefficients (using a setup similar to the one in Example 21.4.8) that will predict the result of every Presidential election from 1788 to the present ("no Democrat has won without state X"). However, there have not been so many Presidential elections since 1788; i.e., not so many equations. So if we incorporate many variables in our model then we will be able to make (meaningless) "predictions" that fit all past elections! A humorous example of the pitfalls of this is given here. ■

21.5. The Rank–Nullity Theorem. At the end of Section 21.3 we discussed an important and very useful result for the study of linear systems, Theorem 21.3.16, a kind of "conservation law" for dimension. To recall the statement of this result and then prove it, we introduce some terminology:

Definition 21.5.1. For an $m \times n$ matrix A, the dimension of $C(A)$ is called the *rank* of A and the dimension of $N(A)$ is called the *nullity* of A.

For instance, A has rank equal to 1 when the column space is 1-dimensional, or in other words, all of its columns lie along a common line through $\mathbf{0}$ (and some column is nonzero). Also, A has positive nullity when its null space contains a nonzero vector.

Example 21.5.2. In Example 21.3.13, the rank of A is 2 and the nullity of A is 1, since $C(A)$ is the span of 2 linearly independent vectors (so its dimension is 2) whereas $N(A)$ is the span of a single nonzero vector (so its dimension is 1). ■

Informally, the rank of A is the number of "degrees of freedom" among the vectors $\mathbf{b} \in \mathbf{R}^m$ for which $A\mathbf{x} = \mathbf{b}$ has *at least one* solution (specifically, $\mathbf{b}$ must belong to the linear subspace $C(A) \subset \mathbf{R}^m$ whose dimension is the rank *by definition*). On the other hand, the nullity of A is a measure of how large the totality of *all* solutions to $A\mathbf{x} = \mathbf{b}$ is provided there is at least one solution (more specifically, if some solution $\mathbf{x}_0$ exists then the solutions are precisely the vectors $\mathbf{x}_0 + \mathbf{v}$ where $\mathbf{v}$ varies in the linear subspace $N(A) \subset \mathbf{R}^n$ whose dimension is the nullity of A *by definition*).

At the end of Section 21.3 we used informal reasoning with "degrees of freedom" and linear transformations to guess the following result, which we shall illustrate and then actually prove.

Theorem 21.5.3 (Rank–Nullity Theorem). If A is an $m \times n$ matrix (so T_A is a linear function $\mathbf{R}^n \to \mathbf{R}^m$) then

$$\mathrm{rank}(A) + \mathrm{nullity}(A) = n \ (= \dim \mathbf{R}^n = \dim(\text{space of inputs for } T_A)).$$

Example 21.5.4. Let us see what the Rank–Nullity Theorem is asserting in some cases that we have already calculated.

For the 2×3 matrix A in Example 21.2.2 we have seen that $C(A) = \mathbf{R}^2$, so the rank is 2. The Rank–Nullity Theorem says that the sum of its rank and nullity is 3, so the nullity must be 1; i.e., the linear subspace $N(A) \subset \mathbf{R}^3$ is a line through the origin. An explicit nonzero vector in $N(A)$ was mentioned in Example 21.4.6, namely $\begin{bmatrix} 1 \\ -14 \\ 4 \end{bmatrix}$, so the Rank–Nullity Theorem says that $N(A)$ is the span of that vector.

For the 3×3 matrix A in Example 21.3.13, we have seen that the rank is 2 and the nullity is 1, and $2 + 1 = 3$ in accordance with the Rank–Nullity Theorem in this case.

For the 4×4 matrix A in Example 21.3.11, we mentioned there that (via methods in Chapter 22) its null space is the span of 2 explicit linearly independent vectors, so its nullity is 2. The Rank–Nullity Theorem then tells us that its rank is $4 - 2 = 2$; i.e., the column space $C(A) \subset \mathbf{R}^4$ is 2-dimensional. This can be seen rather directly as follows. If we denote the first column as $\mathbf{w}$ and the second as $\mathbf{w}'$ then the third column is $2\mathbf{w} - \mathbf{w}'$ and the fourth column is $\mathbf{w} + \mathbf{w}'$ (these can be checked directly; give it a try, and don't worry about how one would discover such relations). Hence, when forming the span of all 4 columns the 3rd and 4th are redundant (they are obtained already from the span of $\mathbf{w}$ and $\mathbf{w}'$), so $C(A) = \mathrm{span}(\mathbf{w}, \mathbf{w}')$. By inspection the columns

$$\mathbf{w} = \begin{bmatrix} 2 \\ -1 \\ 5 \\ 3 \end{bmatrix}, \quad \mathbf{w}' = \begin{bmatrix} 1 \\ 1 \\ -2 \\ 7 \end{bmatrix}$$

are nonzero and not scalar multiples of each other, so they are linearly independent. Hence, these two vectors are a linearly independent spanning set for $C(A)$, so they constitute a basis for $C(A)$; this reaffirms that $\dim C(A) = 2$, which is to say that the rank of A is 2. ■

We now prove the Rank–Nullity Theorem.

PROOF. Our goal is to show that the dimension of $N(A)$ plus the dimension of $C(A)$ must always equal n (which equals the number of columns in A). If A is the $m \times n$ zero matrix then $C(A) = \{\mathbf{0}\}$ and $N(A) = \mathbf{R}^n$, so $\mathrm{rank}(A) = 0$ and $\mathrm{nullity}(A) = n$, confirming the result in this case. Hence, we can assume A is not the $m \times n$ zero matrix, so $C(A)$ is nonzero.

Let $r = \mathrm{rank}(A) = \dim C(A) > 0$, and pick a basis $\mathbf{w}_1, \dots, \mathbf{w}_r$ for $C(A)$. Though there are n columns of A, typically the collection of columns has some "redundancy"; i.e., $r \le n$ and we could certainly have $r < n$. For each $1 \le i \le r$, each $\mathbf{w}_i$ belongs to the column space $C(A) \subset \mathbf{R}^m$ and so by the interpretation of $C(A)$ as the image of T_A we can write $\mathbf{w}_i = A\mathbf{x}_i$ for some $\mathbf{x}_i \in \mathbf{R}^n$ (there may be many solutions to the vector equation $A\mathbf{x} = \mathbf{w}_i$, but just pick one solution $\mathbf{x}_i$ for each i). The collection of $\mathbf{x}_i$'s inherits linear independence from the collection of $\mathbf{w}_i$'s. Indeed, if $\sum_{i=1}^r c_i\mathbf{x}_i = \mathbf{0}$ for some $c_1, \dots, c_r$ then applying T_A to both sides yields $\sum_{i=1}^r c_i\mathbf{w}_i = \mathbf{0}$, which forces all c_i's to vanish due to linear independence of the $\mathbf{w}_i$'s (since they have been chosen as a basis of $C(A)$).

Let $k = \dim N(A)$ denote the nullity of A, so we want to show $r + k = n$. Since all bases of $\mathbf{R}^n$ have size n, to show $r + k = n$ it suffices to build a basis of $\mathbf{R}^n$ consisting of $r + k$ vectors. In case $k = 0$ (i.e., $N(A) = \{\mathbf{0}\}$) we will show that the collection $\{\mathbf{x}_1, \dots, \mathbf{x}_r\}$ of r vectors is a basis of $\mathbf{R}^n$ (so $r = n = n + 0$), and in case $k > 0$ and we pick a basis $\mathbf{y}_1, \dots, \mathbf{y}_k$ for $N(A) \subset \mathbf{R}^n$ then we will show that the collection

$$\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k, \mathbf{x}_1, \dots, \mathbf{x}_r\}$$

of $r + k$ vectors is a basis of $\mathbf{R}^n$.

As a first step, we will show that the desired basis of $\mathbf{R}^n$ is at least a spanning set, and then we will check that it is a linearly independent collection (hence a basis of its span $\mathbf{R}^n$, as desired). To prove the spanning property, pick any $\mathbf{x} \in \mathbf{R}^n$. We know that $A\mathbf{x} \in C(A)$, yet by design $C(A)$ is the span of the $\mathbf{w}_i$'s, so

$$A\mathbf{x} = c_1\mathbf{w}_1 + \cdots + c_r\mathbf{w}_r$$

for some scalars $c_1, \ldots, c_r$. We can rewrite this as

$$A\mathbf{x} = c_1 A\mathbf{x}_1 + \ldots + c_r A\mathbf{x}_r = A(c_1\mathbf{x}_1 + \cdots + c_r\mathbf{x}_r).$$

This shows that both $\mathbf{x}$ and $c_1\mathbf{x}_1 + \ldots + c_r\mathbf{x}_r$ yield the same output in $\mathbf{R}^m$ upon applying A. Passing to the difference as we did early in Section 21.3 yields

$$A(\mathbf{x} - c_1\mathbf{x}_1 - c_2\mathbf{x}_2 - \cdots - c_r\mathbf{x}_r) = A\mathbf{x} - A(c_1\mathbf{x}_1 + \cdots + c_r\mathbf{x}_r) = A\mathbf{x} - (c_1\mathbf{w}_1 + \cdots + c_r\mathbf{w}_r) = \mathbf{0},$$

so by definition of $N(A)$ we have

$$\mathbf{x} - c_1\mathbf{x}_1 - c_2\mathbf{x}_2 - \cdots - c_r\mathbf{x}_r \in N(A). \tag{21.5.1}$$

If $N(A) = \{\mathbf{0}\}$ then this says $\mathbf{x} - c_1\mathbf{x}_1 - \cdots - c_r\mathbf{x}_r = \mathbf{0}$, so $\mathbf{x} = c_1\mathbf{x}_1 + \cdots + c_r\mathbf{x}_r$. This shows $\mathbf{x}$ is in the span of the $\mathbf{x}_j$'s in such cases, as desired. Suppose instead that $N(A)$ is nonzero, so we have chosen a basis $\mathbf{y}_1, \ldots, \mathbf{y}_k$ for the null space. Hence, the vector in (21.5.1) is a linear combination of the $\mathbf{y}_j$'s:

$$\mathbf{x} - c_1\mathbf{x}_1 - c_2\mathbf{x}_2 - \cdots - c_r\mathbf{x}_r = d_1\mathbf{y}_1 + \cdots + d_k\mathbf{y}_k$$

for some scalars $d_1, \ldots, d_k$. This yields that

$$\mathbf{x} = c_1\mathbf{x}_1 + \cdots + c_r\mathbf{x}_r + d_1\mathbf{y}_1 + \cdots + d_k\mathbf{y}_k, \tag{21.5.2}$$

so $\mathbf{x}$ belongs to the collective span of the $\mathbf{x}_i$'s and $\mathbf{y}_j$'s, as desired.

We have established the desired spanning property in general, and it remains to prove the linear independence property. In the case $k = 0$ there is nothing to do because we have already shown that the collection of $\mathbf{x}_i$'s is linearly independent in general. Hence, we can assume $k > 0$, and our aim is to show linear independence of the collection of $\mathbf{x}_i$'s and $\mathbf{y}_j$'s. In other words, if

$$\sum_{i=1}^{r} c_i\mathbf{x}_i + \sum_{j=1}^{k} d_j\mathbf{y}_j = \mathbf{0} \tag{21.5.3}$$

in $\mathbf{R}^n$ then we want to show that all the coefficients c_i and d_j vanish. Applying A to the vanishing sum (21.5.3) and using that $\sum d_j\mathbf{y}_j \in N(A)$ then gives the vanishing of

$$A(\sum c_i\mathbf{x}_i + \sum d_j\mathbf{y}_j) = A(\sum c_i\mathbf{x}_i) + A(\sum d_j\mathbf{y}_j) = A(\sum c_i\mathbf{x}_i) + \mathbf{0} = \sum c_i A(\mathbf{x}_i) = \sum c_i\mathbf{w}_i$$

(we have $A(\sum d_j\mathbf{y}_j) = \mathbf{0}$ since $\sum d_j\mathbf{y}_j \in N(A)$). But the $\mathbf{w}_i$'s are a *basis* for $C(A)$, so the vanishing of $\sum c_i\mathbf{w}_i$ forces all c_i's to vanish.

Since every c_i is equal to 0, (21.5.3) says $\sum d_j\mathbf{y}_j = \mathbf{0}$. But the $\mathbf{y}_j$'s have been chosen as a basis of $N(A)$, so they're a linearly independent collection. Hence, the d_j's all vanish too, so we are done. □

21.6. Applications of Rank–Nullity Theorem: matrix rank and recommender systems (optional). There is an astonishing property of matrices that we do not need in this course but occurs in applications, relates the column space and null space in an essential way, and nicely illustrates the synthesis of geometric and algebraic aspects of linear algebra. We now discuss it and indicate its significance for certain applications.

Recall some terminology from Definition 21.5.1: the dimension of $C(A)$ is called the *rank* of A and is denoted $\mathrm{rank}(A)$. This is the "number of independent columns" in A. Since the n columns $\mathbf{a}_1, \ldots, \mathbf{a}_n$ of A live in $\mathbf{R}^m$, certainly $\mathrm{rank}(A) \leq \dim \mathbf{R}^m = m$.

When $\mathrm{rank}(A) = m$, one says A has "full rank"; this occurs precisely when $C(A) = \mathbf{R}^m$, which is to say that the equation $A\mathbf{x} = \mathbf{b}$ has a solution for every $\mathbf{b} \in \mathbf{R}^m$. At essentially the opposite extreme, $\mathrm{rank}(A) = 1$ when the columns all lie along the same line (and there is some nonzero column). We note that matrices with rank 1, or at least with very small rank, play an essential role in many techniques of applied math such as *rank reduction* and *principal component analysis* that will be discussed in Example 27.3.8 and Example 27.3.11 respectively (and are one of the main tools in certain approaches to image compression).

In certain applications (e.g., Example 21.6.3 below) the rows are of more immediate interest than the columns. We define the *row rank* of A to be the dimension of the span of the rows and may then call the number we defined earlier, $\mathrm{rank}(A)$, the "column rank". Since rows of A become the columns of the transpose $A^\top$, the row rank of A is the same thing as $\mathrm{rank}(A^\top)$.

Now comes the big surprise:

Theorem 21.6.1. For every $m \times n$ matrix A, $\mathrm{rank}(A) = \mathrm{rank}(A^\top)$; i.e., row rank equals column rank!

From the purely algebraic perspective of a matrix being an array of numbers, this equality is an unbelievable miracle. (One of the authors still remembers being stunned upon first encountering this fact as the outcome of a complicated algorithm.) You may even see this equality called the "Fundamental Theorem of Linear Algebra" in other references, but that name is far from universally accepted. The main mathematical point we wish to explain below is that by using our *geometric* view of linear algebra via dimension as a precise version of "degrees of freedom" (the informal idea of "independent" directions of motion), this apparent miracle can be demystified.

Remark 21.6.2. It is sometimes said that the rank of a matrix is its "information content". One reason for this is that if an $m \times n$ matrix A has rank k then to distinguish between two vectors in the span of the rows or two vectors in the span of the columns we only need to make at most k measurements even though the rows live in $\mathbf{R}^n$ and the columns live in $\mathbf{R}^m$ *with m and n possibly vastly bigger than k*. Indeed, granting Theorem 21.6.1, the span of the rows is a k-dimensional subspace of $\mathbf{R}^n$ and the span of the columns is a k-dimensional subspace of $\mathbf{R}^m$, and in general if V is a k-dimensional subspace of $\mathbf{R}^N$ then to describe a vector $\mathbf{v} \in V$ involves specifying its k coefficients as a linear combination of a *fixed* choice of basis of V (which has *nothing* to do with the specific $\mathbf{v} \in V$ under consideration).

Before we carry out the demystification of Theorem 21.6.1, we digress to explain why this result is so useful in applications. Granting such equality, we may unambiguously speak of the "rank" of a matrix: using either rows or columns. Where does the rank (and especially the flexibility to interpret it in terms of either rows or columns) arise in practical applications of linear algebra?

We now give a ubiquitous example, but it is the tip of the iceberg (there are many more, in image compression, search engines, classifiers, video tracking, sensor networks, evolutionary history, etc.).

Example 21.6.3 (Rating matrices). Lots of companies (Amazon, Netflix, Spotify, Apple, etc.) want to recommend future purchases to you (of books, movies, songs, etc.) based on your past purchases or search history. It is very important to these companies that their automated recommendation systems work reasonably well (e.g., the million-dollar Netflix Prize wasn't offered out of a sense of altruism). But how does an automated recommender work, and how is matrix rank (in both the sense of rows *and* the sense of columns) relevant? We now explain the basic idea for movie recommendations in Netflix, but the same circle of ideas applies much more widely. (As Neil Hunt, an early employee of Netflix, once said: "There's a whole lot of Ph.D.-level math and statistics involved" [Aul, p. 58].)

On Netflix, each user can rate a movie they watch with the ratings $1, 2, 3, 4, 5$ (5 = excellent, 1 = very bad). Imagine a "ratings matrix" R whose ij-entry R_{ij} is the rating that the ith user would give to the jth movie *if* every user were to rate every movie. (The rows of R correspond to users, and the columns of R correspond to movies.) Of course, nobody knows the matrix R exactly, because no user ever rates more than a tiny fraction of the huge database of all movies at Netflix. So what Netflix has is rather *limited* knowledge of R, by knowing *some* of the entries thanks to ratings provided by its past users. (Some users never rate any of the movies that they watch, but let's ignore that here.) The goal is to do a good job of filling in the rest of the matrix in order to make recommendations to each user.

There are many thousands of movies in the Netflix system and there are many millions of users, so the number m of rows and number n of columns in R are each *huge*. How can one sensibly fill in the many unknown entries in R? As a purely mathematical problem, it seems hopeless since each missing entry could be pretty much anything from 1 to 5. However, things should not be so random, because the number of "types" of movies isn't remotely on the order of the many thousands of movies that Netflix provides and the number of "types" of people (as far as movie-watching habits are concerned) isn't remotely on the order of many millions of people who are users of Netflix.

So to a very good approximation, the number of actual "degrees of freedom" among the rows (users) and among the columns (movies) should each be rather small (compared to m and n). To make this informal idea of "degrees of freedom" precise, we'll use the concept of dimension in linear algebra: the row space $V = C(R^\top) \subset \mathbf{R}^n$ and the column space $W = C(R) \subset \mathbf{R}^m$ of the $m \times n$ matrix R should each have rather *small* dimensions compared to n and m. (Strictly speaking, what we really expect is that all rows should be rather close to some low-dimensional subspace of $\mathbf{R}^n$, and all columns should be rather close to some low-dimensional subspace of $\mathbf{R}^m$, but for the sake of discussion we'll pretend that V and W are themselves exactly low-dimensional.)

Denote by k the *common* row rank and column rank of R (Theorem 21.6.1!). In the spirit of Remark 21.6.2, we believe that the row vector $\mathbf{v}_i \in V \subset \mathbf{R}^n$ for the ith user is determined by k measurements rather than requiring n measurements. Likewise, we believe that the column vector $\mathbf{w}_j \in W \subset \mathbf{R}^m$ for the jth movie is determined by k measurements rather than requiring m measurements. This idea of "k measurements" is vague, since it entails a choice of basis and there is no particular basis that leaps out at us in either V or W (nor do we even know the value of k).

Now comes a great idea, which shows up all over the place in classifier problems and machine learning more broadly: we *imagine* that all movies are classified according to k categories $C_1, \ldots, C_k$ *which we don't know at all*. This is a classification problem for which neither the actual categories nor even the exact number of them are known! (The role of such categories C_i is solely motivational for us. In other situations with recommender systems, one really wants to *discover* what are such C_i's, including the number k of them; that is the focus of the machine learning technique called "topic modeling".) We will use the idea of such C_i's to "define" hypothetical bases for V and W, which in turn will lead us to a well-posed mathematical strategy to sensibly guess how to fill in the unknown entries in R (at which point the motivation that led to the strategy is no longer logically needed).

The "type" of a movie should be a k-vector whose ath entry p_a (for $1 \leq a \leq k$) is a number between 0 and 1 that measures how well the movie fits into the category C_a. (There might be many p_a's close to 1, and also many close to 0.) Associating to each column $\mathbf{w}_j \in W$ the corresponding k-vector $[\mathbf{w}_j] \in \mathbf{R}^k$ that is the "type" of the jth movie should underlie a specific linear identification of the k-dimensional W with $\mathbf{R}^k$ and thereby encode a natural basis of W (corresponding to the standard basis of $\mathbf{R}^k$). Switching attention to the user side, the "movie taste" of a user should be a k-vector whose ath entry x_a (for $1 \leq a \leq k$) is a number between 1 and 5 that measures how well movies in category C_a are typically rated by that user. Associating to each row $\mathbf{v}_i \in V$ the corresponding

k-vector $[\mathbf{v}_i] \in \mathbf{R}^k$ that is the ith user's "movie taste" should specify a natural basis of V similarly to what we saw with W.

Having now "defined" hypothetical bases for V and W in order to view each as $\mathbf{R}^k$ (via the k coefficients when expressing a vector as a linear combination in the hypothetical basis vectors), it makes sense to form the dot product $[\mathbf{v}_i] \cdot [\mathbf{w}_j] \in \mathbf{R}$ of k-vectors (whereas it makes no sense at all, and certainly not in any way that would ever be useful for our purposes, to define a dot product between general n-vectors $\mathbf{v}$ and m-vectors $\mathbf{w}$). The crucial observation is that, if you think about the *meaning* of x_a and p_a for a minute, this dot product $\sum_{a=1}^{k} x_a p_a$ ought to be the ith user's rating of the jth movie (on the 1-to-5 scale)! That is, we should have $[\mathbf{v}_i] \cdot [\mathbf{w}_j] = R_{ij}$. If we now define M to be the $m \times k$ matrix whose ith row is $[\mathbf{v}_i]$ and N to be the $k \times n$ matrix whose jth column is $[\mathbf{w}_j]$, this says

$$MN = R. \tag{21.6.1}$$

This is all rather informal at the moment: we don't know exactly what k is, nor what the categories C_a are (let alone exactly how to transform row and column data into k-vectors); up to now it is just a theoretical consideration inside our head.

But there is a precise outcome of this informal reasoning: (21.6.1) suggests that whatever R may be, there should be some unknown but not-too-big k, some $m \times k$ matrix M, and some $k \times n$ matrix N for which MN agrees well with R in even the limited range of entries that Netflix *does* know. Precisely because M and N are "low-rank" matrices (N has row rank at most k, and M has column rank at most k), this outcome already turns out to be extremely restrictive on the possibilities for k, M, and N! When combined with an array of advanced statistical methods and a rich body of techniques for working with "low-rank" matrices (a topic we will come back to in Example 27.3.8), it turns out that there is a good way of finding a "best" such k, M, and N. The resulting product MN then provides a reasonable guess for filling in the unknown entries in R. (Moreover, once we have found such k, M, and N, we can use them to make operational definitions of the categories $C_1, \dots, C_k$. This arises in a lot of work on classifier problems.)

We emphasize that for N it is literally the row rank that is small whereas for M it is literally the column rank that is small. Only because of the *equality* of row rank and column rank of a matrix is it unambiguous to speak of a matrix having "low rank", and the intrinsic nature of this concept (not prejudiced towards rows or columns!) is **essential** to the success of algorithms with such matrices. ■

Now let's come back to the task of explaining why the equality of row rank and column rank (Theorem 21.6.1) is much more plausible than it may initially appear to be. Applying the geometrically reasonable Theorem 19.2.5 to the subspace $C(A) \subset \mathbf{R}^m$ shows $\mathrm{rank}(A) = m - \dim C(A)^{\perp}$. Unraveling definitions gives that $C(A)^{\perp}$ is the null space of $A^{\top}$ (please check for yourself; it really involves nothing more than unraveling definitions), so $\mathrm{rank}(A) = m - \dim N(A^{\top})$. Hence, the miraculous-looking assertion $\mathrm{rank}(A) = \mathrm{rank}(A^{\top})$ says equivalently that $\mathrm{rank}(A^{\top}) = m - \dim N(A^{\top})$, or in other words

$$m \overset{?}{=} \dim N(A^{\top}) + \mathrm{rank}(A^{\top})$$

for the $n \times m$ matrix $A^{\top}$. Ah, but this is just the Rank–Nullity Theorem (from Section 21.5) applied to $A^{\top}$!

So the Rank–Nullity Theorem and the geometric Theorem 19.2.5 together contain the entire content of the equality of row rank and column rank. Many references explain the equality of row rank and column rank as an outcome of a long algorithmic process called "row reduction", but the visual approach as above in terms of linear transformations and orthogonality provides a more intuitive understanding.

Chapter 21 highlights (links to highlights in and next chapters)

Notation	Meaning	Location in text
$C(A)$	column space of an $m \times n$ matrix A (subspace of $\mathbf{R}^m$)	Definition 21.2.1
$N(A)$	null space of an $m \times n$ matrix A (subspace of $\mathbf{R}^n$)	Definition 21.3.2
$V \subset W$	V is a subset of W	box above Prop. 21.2.5

Concept	Meaning	Location in text
column space of $m \times n$ matrix A	span in $\mathbf{R}^m$ of the columns in A	Definition 21.2.1
image of linear $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$	all vectors in $\mathbf{R}^m$ obtained as output of $\mathbf{f}$	Definition 21.2.9
null space of $m \times n$ matrix A	all vectors $\mathbf{x} \in \mathbf{R}^n$ for which $A\mathbf{x} = \mathbf{0}$	Definition 21.3.2
overdetermined, *underdetermined* for system of linear equations	more equations than unknowns ("tall" matrix) is *overdetermined*, fewer equations than unknowns ("wide matrix") is *underdetermined*	Definition 21.4.2

Result	Meaning	Location in text
$A\mathbf{x} = \mathbf{b}$ has solution precisely when $\mathbf{Proj}_{C(A)}(\mathbf{b}) = \mathbf{b}$; existence of solution is rare when $C(A) \neq \mathbf{R}^m$	having solution says $\mathbf{b} \in C(A)$, same as $\mathbf{Proj}_{C(A)}(\mathbf{b}) = \mathbf{b}$	Remark 21.2.8
distinct solutions to $A\mathbf{x} = \mathbf{b}$ differ by *nonzero* solution to $A\mathbf{x} = \mathbf{0}$	if $A\mathbf{x}_0 = \mathbf{b} = A\mathbf{x}_1$ then $A(\mathbf{x}_0 - \mathbf{x}_1) = \mathbf{0}$, and $\mathbf{x}_0 - \mathbf{x}_1 \neq \mathbf{0}$ if $\mathbf{x}_0 \neq \mathbf{x}_1$	Example 21.3.1
null space is a linear subspace	though defined by a system of equations (admitting $\mathbf{0}$ as a solution), it is span of finitely many solutions	Proposition 21.3.6
linear subspaces of $\mathbf{R}^n$ are precisely the collections V of vectors for which: $\mathbf{0} \in V$ and any linear combination of vectors in V is in V	characterize linear subspaces in terms of *properties* rather than a description (as a span)	Theorem 21.3.7
general solution to $A\mathbf{x} = \mathbf{b}$ is built from a particular solution $\mathbf{x}_0$ and $N(A)$	describe all solutions in terms of $\mathbf{x}_0$ and solutions to $A\mathbf{x} = \mathbf{0}$	Prop. 21.3.10, Thm. 21.3.14
Rank–Nullity Thm (a "conservation law" for dimension)	interpreting $A\mathbf{x} = \mathbf{b}$ in terms of $T_A : \mathbf{R}^n \to \mathbf{R}^m$, compares dimension of output and input in terms of $N(A)$	Ex. 21.3.15, Thm. 21.3.16

Skill	Location in text
understand why $A\mathbf{x} = \mathbf{b}$ has a solution precisely when $\mathbf{b} \in C(A)$	(21.2.4)
compute $\mathbf{Proj}_{C(A)}(\mathbf{b})$ to check if $A\mathbf{x} = \mathbf{b}$ has a solution	Example 21.2.7
visualize image of linear transformation	Examples 21.2.10, 21.2.11
from particular solution and basis of $N(A)$, describe all solutions to $A\mathbf{x} = \mathbf{b}$ in parametric form	Example 21.3.13
visualize the ideas of overdetermined and underdetermined	Figure 21.4.1, Figure 21.4.2, Remark 21.4.5, Example 21.4.6.

22. Matrix decompositions: QR-decomposition and LU-decomposition

In this chapter we describe an approach to solving a linear system $A\mathbf{x} = \mathbf{b}$ that involves writing A as a product of simpler matrices. There are many such *matrix decompositions* (or *matrix factorizations*, as they are sometimes called) that are useful for this and related problems; we discuss two of them, called the QR-decomposition and LU-decomposition. For these methods, the first key point is to express an arbitrary matrix A as a product $A = QR$ for Q with orthonormal columns and R upper triangular (also called "right triangular"), or as a product $A = LU$ for lower triangular L and upper triangular U. The second key point is to understand why such expressions for A help with solving $A\mathbf{x} = \mathbf{b}$.

Matrices have been recognized as important mathematical objects since the 19th century (see [**Ha1**] for an interesting historical survey), but the use of matrix decompositions to solve linear systems efficiently and accurately started only in the 1950's. On electronic computers of the mid-1940's, the task of accurately solving (say to 6 decimal digits) a general linear system of 10 equations in 10 unknowns was regarded as essentially hopeless, due to accumulated rounding errors. J.H. Wilkinson[19], who worked with Turing in those days, had the following to say in his acceptance speech for the 1970 Turing Award (see [**Wil2**, p. 143]).

> *When I joined [the National Physical Laboratory] in 1946 the mood of pessimism about the stability of elimination methods for solving linear systems [of n equations in n unknowns] was at its height and was a major talking point. Bounds had been produced which purported to show that the error in the solution would be proportional to 4^n and this suggested that it would be impractical to solve systems even of quite modest order. I think it was true to say that at that time (1946) it was the more distinguished mathematicians who were most pessimistic, the less gifted being perhaps unable to appreciate the full severity of the difficulties.*

By the mid-1960's, the difficulties were mostly overcome using an array of matrix decompositions. The moral of the story is: the leading "more distinguished" experts can sometimes be too pessimistic!

Such matrix decompositions for a given matrix A will be used to solve $A\mathbf{x} = \mathbf{b}$ fairly efficiently. The QR-decomposition expresses the outcome of the Gram–Schmidt process from Chapter 19 applied to the columns of A, and it *arises when solving many other problems with real-world significance* (see Section 22.5, especially from Proposition 22.5.1 onwards, and Appendix H)! The LU-decomposition is essentially the outcome of "row reduction" (also called Gaussian elimination), which for 2×2 matrices A is a repackaging of the usual algebra method to solve 2 linear equations in 2 unknowns.

By the end of this chapter you should be able to:

- solve a linear system or invert a matrix, given a suitable matrix decomposition (QR or LU);
- use Gram–Schmidt to construct a QR-decomposition.

These two decompositions are far from the end of the story: efficient implementation of them on a computer requires further work. Since these (and their generalizations, as well as other important decompositions) are so important for numerical work, many clever modifications have been made which optimize their use and implementation. This chapter does not treat any of that more advanced material; we are just giving an introduction to the idea and applications of matrix decompositions as an illustration of the power of matrix algebra. Our emphasis is on how to use these decompositions (including purposes beyond solving linear systems!). There are many software tools to find the decompositions for a given matrix, and the range of applications of matrix decompositions is truly vast.

---

[19] James H. Wilkinson (1919-1986) was a British mathematician who made fundamental contributions to numerical analysis and the development of accurate and efficient computational algorithms on computers, especially centered around linear algebra. He was on the Stanford CS faculty (not math faculty, oddly enough) during 1977-1984, but only in residence during the winter.

22.1. Solving some special linear systems. We have already discussed at length in Chapter 21 what to expect concerning whether a given linear system $A\mathbf{x} = \mathbf{b}$ of m equations in n unknowns has a solution, and if so then how many. However, when faced with a big linear system in a real-world problem, we need to either (i) find the solution(s) numerically, or (ii) find a good "approximate solution" if there is no exact solution. The matrices arising in contemporary practical applications can be quite large, sometimes with the number of equations and unknowns each on the order of millions or billions, and (to make matters worse) the matrix entries are rarely whole numbers or simple fractions. This puts a huge emphasis on methods that are efficient, in the sense of minimizing the number of actual arithmetic calculations that need to be performed, and approximate the solution(s) to a high degree of accuracy.

When grappling with very large linear systems (or really, anything with $m, n > 3$), you will undoubtedly want to use some computational software rather than compute the answer by hand. *All computer-assisted methods rely in an essential way on matrix algebra* in the sense that they all involve the determination and use of a *matrix decomposition*: writing a matrix A as a product of "simpler" matrices and leveraging special properties of those simpler matrices to find the solution in stages.

If you look at the Wikipedia page for "matrix decomposition", you will find many such methods. We shall discuss two of the most widely used (called the LU and QR methods, for reasons to be explained). It is important to keep in mind several points:

(i) There is no overall *best* method; many different matrix decompositions exist, and the choice of which one to use depends on the task at hand. Although we focus on the LU and QR-decompositions, there are other decompositions which are better suited for handling certain types of matrices that arise in some specific applications.

(ii) We discuss here only the most bare-bones version of the LU and QR methods, but in practice with large linear systems it is crucial to incorporate refinements to speed up the processing time and improve the numerical stability.

(iii) For solving linear systems of n equations in n unknowns, the LU method is faster than the QR method by a factor of 2; that difference mattered a lot back when the time comparison was 10 hours versus 5 hours; nowadays it is often a difference of milliseconds and so perhaps is not so crucial anymore. However, more significantly, the LU method has the defect that for certain matrices an LU-decomposition may not exist. Fortunately, such "bad" matrices are rare (and the LU method can be adjusted to handle them). However, it is very difficult to predict beforehand whether a matrix that arises for any particular application is one of these problematic matrices.

Matrix decompositions and their application to solving $A\mathbf{x} = \mathbf{b}$ for general A are discussed in Section 22.2 for square A and Section 22.5 for general A. In this first section, as a warm-up we'll see that for special matrices A the linear system $A\mathbf{x} = \mathbf{b}$ is very simple to solve. These examples might look too special to be of general interest, but miraculously the opposite is true: we shall build upon these special cases to tackle $A\mathbf{x} = \mathbf{b}$ for general A!

Example 22.1.1. Suppose that A is an upper triangular square matrix; this means that it is an $n \times n$ matrix for which every entry below the diagonal vanishes; i.e., $a_{ij} = 0$ whenever $i > j$. (We have encountered this earlier, in Example 15.1.7.) If we write out the corresponding linear system, it might look like this:

$$\begin{aligned} 3x_1 - 2x_2 + 4x_3 - 8x_4 &= 32 \\ 9x_2 + 5x_3 + 6x_4 &= -3 \\ x_3 + 2x_4 &= -1 \\ -5x_4 &= 25 \end{aligned}$$

We will proceed by working our way up from the bottom. First solve the last equation: $x_4 = -5$. Now substitute this into the next to last equation to get $x_3 + 2(-5) = -1$, or in other words $x_3 = 9$. Then substitute these two values for x_4 and x_3 into the third to last equation, and so on. (One gets $x_2 = -2$, and finally $x_1 = -16$ via the first equation.)

This method is called *back-substitution*. It is quite easy to carry out to uniquely solve $A\mathbf{x} = \mathbf{b}$ for any $n \times n$ upper triangular A and n-vector $\mathbf{b}$ *provided* that the diagonal entries of the upper triangular square matrix A are nonzero. The non-vanishing of the diagonal entries corresponds to the coefficient of x_j in the jth equation being nonzero, which is necessary in order to (uniquely) solve for x_j at that step.

Essentially the same calculations also work for lower triangular square matrices A (i.e., those whose entries above the diagonal all vanish), except that we work our way from the top to the bottom: one solves the first equation for x_1, then the second equation for x_2, and so on. For example, in the linear system

$$\begin{aligned} 2x_1 \qquad\qquad\quad &= 6 \\ -3x_1 + x_2 \qquad\quad &= -7 \\ x_1 + 4x_2 + 3x_3 &= 5 \end{aligned} \tag{22.1.1}$$

the first equation tells us $x_1 = 3$, and plugging that into the second equation gives $-9 + x_2 = -7$, so $x_2 = 2$. Plugging these values for x_1 and x_2 into the third equation gives $3 + 8 + 3x_3 = 5$, which says $11 + 3x_3 = 5$, so $3x_3 = -6$ and hence $x_3 = -2$. Putting it all together gives the unique solution $(x_1, x_2, x_3) = (3, 2, -2)$. ■

Remark 22.1.2. Hopefully the specific numerical illustrations in Example 22.1.1 convinced you that back-substitution *always* works for upper triangular and lower triangular square matrices A when all a_{jj} are nonzero, but let's now discuss why it really works for such general upper triangular $n \times n$ matrices A (the lower triangular case goes similarly). The last equation $a_{nn}x_n = b_n$ in an upper triangular system has the unique solution $x_n = b_n/a_{nn}$, and at the stage of solving for x_j we have already uniquely determined the values of $x_{j+1}, \dots, x_n$, so upon inserting those values into the jth equation we uniquely solve for x_j by dividing throughout by its coefficient a_{jj}.

This division step is where we could run into problems if $a_{jj} = 0$. (It is shown in Remark 22.6.3 that an upper triangular $n \times n$ matrix A with a vanishing diagonal entry really is *never* invertible: its null space $N(A)$ is always nonzero and its column space $C(A)$ inside $\mathbf{R}^n$ never coincides with $\mathbf{R}^n$.) For example, we get into trouble if we find ourselves trying to solve the upper triangular system

$$\begin{aligned} 3x_1 - 2x_2 + 4x_3 &= 6 \\ 0x_2 + 5x_3 &= -3 \\ 2x_3 &= -4. \end{aligned}$$

The last equation can be used to determine x_3, but then the second-to-last equation produces a conflict which cannot be resolved by choosing the value of x_2 appropriately. (If the right side of the second equation is anything other than -10 then there is such a conflict.)

Since $A\mathbf{x} = \mathbf{b}$ is uniquely solvable for any $\mathbf{b}$ when A is an upper or lower triangular square matrix with all diagonal entries nonzero, Proposition 18.1.5 and Definition 18.1.6 yield:

Theorem 22.1.3. If A is an upper or lower triangular $n \times n$ matrix with all diagonal entries nonzero then A is invertible.

Next suppose A is an $m \times n$ matrix with orthonormal columns. (For $m = n = 3$ these are the rotation matrices up to an overall sign; see Remark E.5.3.) To solve $A\mathbf{x} = \mathbf{b}$ we try multiplying each side by $A^\top$ on the left since $A^\top A = \mathrm{I}_n$ by orthonormality of the columns (Theorem 20.4.1):

$$\text{if } A\mathbf{x} = \mathbf{b} \text{ then } A^\top(A\mathbf{x}) = A^\top\mathbf{b}, \text{ and } A^\top(A\mathbf{x}) = (A^\top A)\mathbf{x} = \mathrm{I}_n\mathbf{x} = \mathbf{x}, \text{ so } \mathbf{x} = A^\top\mathbf{b}.$$

Hence $\mathbf{x} = A^\top \mathbf{b}$ is the only *possible* solution to $A\mathbf{x} = \mathbf{b}$, and if $m = n$ then it really is a solution because

$$\mathbf{x} = A^\top \mathbf{b} \text{ implies } A\mathbf{x} = A(A^\top \mathbf{b}) = (AA^\top)\mathbf{b} = \mathrm{I}_n \mathbf{b} = \mathbf{b}$$

(this used that $AA^\top = \mathrm{I}_n$ when $m = n$, by Theorem 20.4.1). We have shown:

Theorem 22.1.4. For an $n \times n$ orthogonal matrix A, the equation $A\mathbf{x} = \mathbf{b}$ has exactly one solution: $\mathbf{x} = A^\top \mathbf{b}$. (The optional Section 22.5 discusses an analogue for $m \times n$ matrices A having orthonormal columns with any $m \geq n$, and illustrates its great utility.)

In practice one often encounters "non-square" linear systems $A\mathbf{x} = \mathbf{b}$; i.e., an $m \times n$ matrix A with $m \neq n$ ("number of equations" not equal to "number of unknowns"). The simplest of these to handle will be A that are upper triangular or lower triangular in the sense introduced in Example 15.1.7 that we now review. For a general $m \times n$ matrix with $m \neq n$, to visualize the meaning of "upper triangular" and "lower triangular" there are two cases for each, depending on whether $m < n$ or $m > n$. This is illustrated schematically as follows: in the underdetermined case ($m < n$: fewer equations than unknowns)

$$U = \begin{bmatrix} * & * & \dots & * & * & \dots & * \\ 0 & * & \dots & * & * & \dots & * \\ \vdots & \vdots & \ddots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & * & * & \dots & * \\ 0 & 0 & \dots & 0 & * & \dots & * \end{bmatrix}, \quad L = \begin{bmatrix} * & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ * & * & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \dots & \vdots \\ * & * & \dots & * & 0 & 0 & \dots & 0 \\ * & * & \dots & * & * & 0 & \dots & 0 \end{bmatrix} \tag{22.1.2}$$

and in the overdetermined case ($m > n$: more equations than unknowns)

$$U = \begin{bmatrix} * & * & \dots & * & * \\ 0 & * & \dots & * & * \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & * & * \\ 0 & 0 & \dots & 0 & * \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix}, \quad L = \begin{bmatrix} * & 0 & \dots & 0 & 0 \\ * & * & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ * & * & \dots & * & 0 \\ * & * & \dots & * & * \\ \vdots & \vdots & \dots & \vdots & \vdots \\ * & * & \dots & * & * \end{bmatrix} \tag{22.1.3}$$

with upper triangular U and lower triangular L. The next four examples solve systems of each type above.

Example 22.1.5 (underdetermined upper triangular). Consider the underdetermined linear system of 3 equations in 5 unknowns

$$\begin{aligned} 2x_1 - 3x_2 + x_3 + 3x_4 - x_5 &= 2 \\ x_2 + 4x_3 + 4x_4 - 7x_5 &= 5 \\ 3x_3 + 3x_4 - 6x_5 &= 3 \end{aligned}$$

This has the form $U\mathbf{x} = \mathbf{b}$ with $U = \begin{bmatrix} 2 & -3 & 1 & 3 & -1 \\ 0 & 1 & 4 & 4 & -7 \\ 0 & 0 & 3 & 3 & -6 \end{bmatrix}$ upper triangular as on the left in (22.1.2). Because U has nonzero diagonal entries, this will be easy to solve via back-substitution; the "rule of thumb" from Section 21.4 that there are infinitely many solutions will also be verified explicitly.

The key insight is that the "extra variables" x_4 and x_5 corresponding to the "extra columns" of U (i.e., the columns beyond those meeting the diagonal of U) can be moved to the right side of each equation, turning this into an upper triangular *square* system (of 3 equations in 3 unknowns)

$$\begin{aligned} 2x_1 - 3x_2 + x_3 &= 2 - 3x_4 + x_5 \\ x_2 + 4x_3 &= 5 - 4x_4 + 7x_5 \\ 3x_3 &= 3 - 3x_4 + 6x_5 \end{aligned} \tag{22.1.4}$$

upon *choosing* values of x_4 and x_5 at will. For instance, if we choose $x_4 = 2$ and $x_5 = -1$ then this becomes the *square* upper triangular system

$$\begin{aligned} 2x_1 - 3x_2 + x_3 &= -5 \\ x_2 + 4x_3 &= -10 \\ 3x_3 &= -9 \end{aligned} \tag{22.1.5}$$

We solve for the remaining variables x_1, x_2, x_3 by back-substitution: $x_3 = -3$, $x_2 = 2$, $x_1 = 2$ (with $x_4 = 2, x_5 = -1$). Please do the back-substitution yourself as a check on your understanding.

The same process works for any specific values of x_4 and x_5, leading us to realize it even works treating x_4 and x_5 symbolically by solving (22.1.4) successively for x_3, x_2, x_1 in terms of x_4 and x_5 via back-substitution (doing the same algebra with symbolic x_4 and x_5 that we did above with (22.1.5) for the numerical values $x_4 = 2$, $x_5 = -1$): we obtain $x_3 = 1 - x_4 + 2x_5$, then $x_2 = 1 - x_5$, and finally $x_1 = 2 - x_4 - 2x_5$. (Please do the back-substitution yourself as a check on your understanding.) This says

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 2 - x_4 - 2x_5 \\ 1 - x_5 \\ 1 - x_4 + 2x_5 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + x_4 \begin{bmatrix} -1 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix} + x_5 \begin{bmatrix} -2 \\ -1 \\ 2 \\ 0 \\ 1 \end{bmatrix}. \tag{22.1.6}$$

In the expression on the right, the first vector $\begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ is a particular solution of the initial system $U\mathbf{x} = \begin{bmatrix} 2 \\ 5 \\ 3 \end{bmatrix}$ (corresponding to the values $x_4 = 0$ and $x_5 = 0$). The vectors multiplying x_4 and x_5 on the right in (22.1.6) constitute a basis for the null space $N(U)$ since vectors in $N(U)$ precisely account for the "degrees of freedom" in the collection of *all* solutions to $U\mathbf{x} = \mathbf{b}$ for a given $\mathbf{b}$, as in Theorem 21.3.14(iii). There are two "degrees of freedom" when solving the system, reflecting that x_4 and x_5 may be chosen arbitrarily and corresponding to the fact that $\dim N(U) = 2$.

The procedure works the same way for any $\mathbf{b}$ whatsoever, so $U\mathbf{x} = \mathbf{b}$ has a solution for any $\mathbf{b}$ (hence $C(U) = \mathbf{R}^3$) and the solutions can be described with "two degrees of freedom" (expressed in terms of x_4 and x_5, making Theorem 21.3.14(iii) explicit). ■

Example 22.1.6 (underdetermined lower triangular). The underdetermined lower triangular case is much easier (and so less interesting) than the underdetermined upper triangular case: it is a linear system whose "extra variables" have coefficients equal to 0 and so don't appear at all! Consider $L\mathbf{x} = \begin{bmatrix} 6 \\ -7 \\ 5 \end{bmatrix}$ for the lower triangular 3×5 matrix $L = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ -3 & 1 & 0 & 0 & 0 \\ 1 & 4 & 3 & 0 & 0 \end{bmatrix}$ (corresponding to the schematic on the right in (22.1.2)) and unknown $\mathbf{x} \in \mathbf{R}^5$. When we compute the 3-vector $L\mathbf{x}$, the variables x_4 and x_5 never appear in the output since the 4th and 5th columns of L vanish. Thus, the linear system $L\mathbf{x} = \begin{bmatrix} 6 \\ -7 \\ 5 \end{bmatrix}$ with unknown $\mathbf{x} \in \mathbf{R}^5$ is our old friend (22.1.1) but viewed as equations on $\mathbf{R}^5$ that don't involve x_4 or x_5. This is akin to regarding "$2x_1 - 3x_2 = 8$" as an equation on $\mathbf{R}^3$ not involving x_3 (defining a plane parallel to the x_3-axis) rather than on $\mathbf{R}^2$ (defining a line).

Doing back-substitution from the top to bottom uniquely determines $x_1 = 3$, $x_2 = 2$, $x_3 = -2$, and there are no constraints on x_4 or x_5 (they don't even appear in the equations!), yielding as the solutions

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ -2 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ -2 \\ 0 \\ 0 \end{bmatrix} + x_4 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + x_5 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Once again the first vector on the right side corresponds to a particular solution (with $x_4 = 0, x_5 = 0$), this one being the unique solution to the corresponding 3×3 square system that ignores the invisible variables x_4 and x_5, and there are again two degrees of freedom corresponding to x_4 and x_5 that can be assigned whatever values we want. In this case $N(L)$ has a basis $\{\mathbf{e}_4, \mathbf{e}_5\}$ consisting of the standard basis vectors corresponding to the invisible variables x_4, x_5. ■

Next we consider some *overdetermined* $m \times n$ triangular systems $A\mathbf{x} = \mathbf{b}$. Since $C(A)$ is the subspace of $\mathbf{R}^m$ spanned by the n columns of A, so $\dim C(A) \le n < m$, for dimension reasons $C(A)$ cannot coincide with $\mathbf{R}^m$. Thus, a typical $\mathbf{b} \in \mathbf{R}^m$ does not belong to $C(A)$ (e.g., for $m = 3$ the subspace $C(A)$ may be a line or plane in $\mathbf{R}^3$ through $\mathbf{0}$, which misses most points $\mathbf{b} \in \mathbf{R}^3$). In other words, $A\mathbf{x} = \mathbf{b}$ has no solution unless $\mathbf{b}$ is quite special (depending on A), so to organize our work we will consider a specific A but leave $\mathbf{b}$ in a form with symbolic entries b_j that we regard as numbers for which we seek the *constraints* on them ensuring $A\mathbf{x} = \mathbf{b}$ has a solution (i.e., $\mathbf{b} \in C(A)$).

Example 22.1.7 (overdetermined upper triangular). Consider the linear system $U\mathbf{x} = \mathbf{b}$ of 5 equations in 3 unknowns with the upper triangular 5×3 matrix $U = \begin{bmatrix} 2 & -3 & 1 \\ 0 & 1 & 4 \\ 0 & 0 & 3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$; written out, it says

$$\begin{aligned} 2x_1 - 3x_2 + x_3 &= b_1 \\ x_2 + 4x_3 &= b_2 \\ 3x_3 &= b_3 \\ 0 &= b_4 \\ 0 &= b_5 \end{aligned} \tag{22.1.7}$$

for unknowns x_1, x_2, x_3, and b_j's that we imagine to be specific numbers. We consider all possibilities for $\mathbf{b}$ because it will clarify our work to treat the b_j's symbolically (but we think of them as given numbers).

The last 2 equations in (22.1.7), corresponding to the "extra rows" of U, do not involve any of the variables; this always happens with the extra rows at the bottom in overdetermined upper triangular cases, and it tells us that there is no solution unless $b_4 = 0$ and $b_5 = 0$. This makes explicit that the column space $C(U)$ of the 5×3 upper triangular U is *not* all of $\mathbf{R}^5$ (it is spanned by the 3 columns). We shall next show that $C(U)$ consists of precisely those $\mathbf{b} \in \mathbf{R}^5$ whose last two entries b_4 and b_5 equal 0.

If $b_4 = 0$ and $b_5 = 0$ then the last 2 equations in (22.1.7) tell us nothing, so we can ignore them and the remaining 3 equations constitute an upper triangular *square* system in the variables x_1, x_2, x_3 that we can solve by back-substitution. This is the square system we dealt with in Example 22.1.5 (upon specifying values for x_4 and x_5 there), so we solve it as we did for (22.1.5), always getting exactly one solution. The uniqueness of the solution tells us $N(U) = \{\mathbf{0}\}$, by Theorem 21.3.14. ■

Example 22.1.8 (overdetermined lower triangular). Finally, consider the linear system $L\mathbf{x} = \mathbf{b}$ of 5 equations in 3 unknowns for the lower triangular 5×3 matrix $L = \begin{bmatrix} 2 & 0 & 0 \\ -3 & 1 & 0 \\ 1 & -4 & 3 \\ 2 & -7 & 3 \\ 4 & 5 & -1 \end{bmatrix}$; written out, it says

$$\begin{aligned} 2x_1 \qquad\qquad &= b_1 \\ -3x_1 + x_2 \qquad &= b_2 \\ x_1 - 4x_2 + 3x_3 &= b_3 \\ 2x_1 - 7x_2 + 3x_3 &= b_4 \\ 4x_1 + 5x_2 - x_3 &= b_5 \end{aligned} \tag{22.1.8}$$

for unknowns x_1, x_2, x_3, and b_j's that we imagine to be specific numbers. As in Example 22.1.7, it will clarify our work to treat the b_j's symbolically (even though we think of them as given numbers).

Focus on the first 3 equations in (22.1.8), ignoring the last 2 equations (corresponding to the "extra rows" in L). This is a lower triangular *square* system for the unknown $\mathbf{x} \in \mathbf{R}^3$. We know how to uniquely solve such square systems via back-substitution. Doing this by treating b_1, b_2, b_3 symbolically, some care with the algebra yields (please try for yourself)

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} (1/2)b_1 \\ (3/2)b_1 + b_2 \\ (11/6)b_1 + (4/3)b_2 + (1/3)b_3 \end{bmatrix}. \tag{22.1.9}$$

(As a safety check, if we plug this into the left side of the first 3 equations in (22.1.8) we indeed get b_1, b_2, b_3 as expected.) For instance, if $(b_1, b_2, b_3) = (6, -7, -11)$ then (22.1.9) says that the *square* system given by the first 3 equations in (22.1.8) has unique solution $\mathbf{x} = (3, 2, -2)$.

Having found a unique $\mathbf{x} \in \mathbf{R}^3$ as in (22.1.9) via the first 3 equations in (22.1.8), the remaining 2 equations at the bottom of (22.1.8) (corresponding to the "extra rows" in L) do not involve anything unknown! That is, we now know what every x_i *must* be, so solving (22.1.8) amounts to checking if those specific x_i's also satisfy the 2 equations at the bottom of (22.1.8); these are constraints on b_4 and b_5.

To illustrate this, again suppose $(b_1, b_2, b_3) = (6, -7, -11)$, so we have seen that the lower triangular square system consisting of the first 3 equations in (22.1.8) has the unique solution $\mathbf{x} = (3, 2, -2)$. Plugging that into the overdetermined system (22.1.8), the final 2 equations then say

$$2(3) - 7(2) + 3(-2) = b_4, \quad 4(3) + 5(2) - 1(-2) = b_5;$$

i.e., $(b_4, b_5) = (-14, 24)$. Hence, for $\mathbf{b} = (6, -7, -11, b_4, b_5)$ the system $L\mathbf{x} = \mathbf{b}$ has the unique solution $\mathbf{x} = (3, 2, -2) \in \mathbf{R}^3$ when $(b_4, b_5) = (-14, 24)$ and *no solution* when $(b_4, b_5) \neq (-14, 24)$.

In general, being careful with the algebra, plugging (22.1.9) in for the x_i's on the left in the last 2 equations in (22.1.8) turns those equations into

$$-4b_1 - 3b_2 + b_3 = b_4, \quad (23/3)b_1 + (11/3)b_2 - (1/3)b_3 = b_5.$$

These two equations on the b_j's characterize the 5-vectors $\mathbf{b}$ for which the overdetermined system $L\mathbf{x} = \mathbf{b}$ has a solution $\mathbf{x} \in \mathbf{R}^3$, so they describe the subspace $C(L)$ inside $\mathbf{R}^5$, and $N(L) = \{\mathbf{0}\}$ by Theorem 21.3.14 since we have seen that a solution to $L\mathbf{x} = \mathbf{b}$ is unique *when it exists*. ■

We now record general conclusions that express the lessons illustrated in Examples 22.1.5–22.1.8.

Theorem 22.1.9. Let A be an upper or lower triangular $m \times n$ matrix *with all diagonal entries nonzero*.

(i) If $m = n$ then $A\mathbf{x} = \mathbf{b}$ always has a unique solution (found via back-substitution).

(ii) In the underdetermined case ($m < n$), $A\mathbf{x} = \mathbf{b}$ always has a solution (i.e., $C(A) = \mathbf{R}^m$) and $\dim N(A) = n - m > 0$. The solutions are given by uniquely solving for $x_1, \ldots, x_m$ in terms of arbitrarily chosen $x_{m+1}, \ldots, x_n$. For lower triangular A, the linear system $A\mathbf{x} = \mathbf{b}$ is really m equations in $x_1, \ldots, x_m$ with the other x_j's having coefficient 0 in every equation.

(iii) In the overdetermined case ($m > n$), $\dim C(A) = n < m$ (so $A\mathbf{x} = \mathbf{b}$ only has a solution for special $\mathbf{b}$) and $N(A) = \{\mathbf{0}\}$ (so a solution to $A\mathbf{x} = \mathbf{b}$ is unique when one exists). The column space $C(A)$ is described by expressions for each of $b_{n+1}, \ldots, b_m$ in terms of $b_1, \ldots, b_n$; for upper triangular A these expressions are simply $b_{n+1} = 0, \ldots, b_m = 0$.

22.2. Solving square linear systems via matrix decompositions. We have now learned how to solve $A\mathbf{x} = \mathbf{b}$ when A is orthogonal or is upper or lower triangular with all diagonal entries nonzero. This might seem too special to be of much use, but it is a remarkable fact that (up to some minor adjustments we shall discuss) for any matrix A we have (away from rare situations) $A = LU$ where L is lower triangular and U is upper triangular and also (with no exceptions) $A = QR$ where Q has orthonormal columns and R is upper triangular. These will enable us to solve $A\mathbf{x} = \mathbf{b}$ by leveraging the special cases from Section 22.1.

To focus on one thing at a time, in this section we concentrate on *square* matrices A; the situation for non-square A (beyond triangular cases) is discussed in the optional Section 22.5.

Theorem 22.2.1.

(i) (LU-decomposition) "Most" $n \times n$ matrices A have the form $A = LU$ for $n \times n$ lower triangular L and $n \times n$ upper triangular U. The matrix A is invertible precisely when the diagonal entries of L and U are nonzero.

(ii) (QR-decomposition) An invertible $n \times n$ matrix A can be written as $A = QR$ where Q is an $n \times n$ orthogonal matrix and R is an $n \times n$ upper triangular matrix with positive diagonal entries.

Remark 22.2.2. In Theorem 22.2.1(i), the meaning of "most" is that only those A whose entries satisfy certain special conditions fail to admit an LU-decomposition. For such special A, an LU-decomposition exists after rearranging the rows of A; this is illustrated in Remark 22.6.2. Theorem 22.2.1(ii) has a formulation that works without assuming invertibility, but we won't discuss it in homework or exams (since this is just a first course), so you can focus on invertible A for QR-decompositions. But going beyond the invertible (and beyond the square) case for the QR-decomposition is very important in practical applications, as is discussed and illustrated at length in Section 22.5 and in Math 104.

In case you are curious, the version of Theorem 22.2.1(ii) that works for any nonzero $n \times n$ matrix A is that $A = QR$ where Q is an $n \times k$ matrix with orthonormal columns for $k = \dim C(A)$ ($\leq n$) and R is a $k \times n$ upper triangular matrix with nonnegative diagonal entries. The matrix A is invertible precisely when $k = n$ (in which case Q is an orthogonal matrix) and the nonnegative diagonal entries of R are nonzero (i.e., positive). An illustration with a non-invertible 3×3 matrix A and $k = 2$ is given in Example 22.4.5.

In the remainder of this section we explain how such decompositions are used to solve $A\mathbf{x} = \mathbf{b}$ where A is $n \times n$ and *we are given* the factors (either L and U, or Q and R). We do not yet address the question of how to construct either of these decompositions of A; in Section 22.6 we sketch how to find an LU-decomposition (when it exists), and in Section 22.4 it will be seen that finding a QR-decomposition is tantamount to carrying out the Gram–Schmidt process from Section 19.2 for the columns of A. Computer software is very well-adapted to compute both types of decompositions (see Remark 22.2.6).

Suppose we have factored a given $n \times n$ matrix A as LU for $n \times n$ lower triangular L and $n \times n$ upper triangular U with all diagonal entries nonzero. To solve $A\mathbf{x} = \mathbf{b}$, we use that $A\mathbf{x} = (LU)\mathbf{x} = L(U\mathbf{x})$ to rewrite the system as $L(U\mathbf{x}) = \mathbf{b}$ and so break the task into two separate problems:

$$\text{solve } L\mathbf{y} = \mathbf{b}, \quad \text{and then solve } U\mathbf{x} = \mathbf{y}.$$

In other words, we introduce an "intermediate" unknown n-vector $\mathbf{y}$ and solve the equation $L\mathbf{y} = \mathbf{b}$ for this vector, and then use such $\mathbf{y}$ as the right side in the equation $U\mathbf{x} = \mathbf{y}$ (so $A\mathbf{x} = L(U\mathbf{x}) = L\mathbf{y} = \mathbf{b}$).

The gain in turning the original $n \times n$ system into a pair of such systems is that these two new intermediate systems are of the much easier type that we learned how to solve in Section 22.1! We solve $L\mathbf{y} = \mathbf{b}$ by back-substitution as in Example 22.1.1 since L is $n \times n$ lower triangular with all diagonal entries nonzero, and similarly for $U\mathbf{x} = \mathbf{y}$ since U is $n \times n$ upper triangular with all diagonal entries nonzero. For the LU-decomposition of a general (perhaps non-invertible) $n \times n$ matrix, either L or U may have some 0's on its diagonal. These possibilities cause extra complications (as seen in Remark 22.1.2) that we avoid here.

Example 22.2.3. Suppose we are given that $A = \begin{bmatrix} 5 & 0 & 1 \\ 1 & 2 & 2 \\ -1 & 4 & 2 \end{bmatrix}$ can be factored as

$$A = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1/5 & 1 & 0 \\ -1/5 & 2 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 5 & 0 & 1 \\ 0 & 2 & 9/5 \\ 0 & 0 & -7/5 \end{bmatrix}}_{U}.$$

Finding such a factorization requires work that is safest to relegate to a computer, but once you see this decomposition you can multiply the matrices to check that it is actually true that A is the product of this L and U. We now use it to solve $A\mathbf{x} = \mathbf{b} = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}$.

Rewrite "$A\mathbf{x} = \mathbf{b}$" as $(LU)\mathbf{x} = \mathbf{b}$ or equivalently $L(U\mathbf{x}) = \mathbf{b}$. Now we set $\mathbf{y} = U\mathbf{x}$ to break our problem into two steps: solve $L\mathbf{y} = \mathbf{b}$ and then solve $U\mathbf{x} = \mathbf{y}$ (so $A\mathbf{x} = L(U\mathbf{x}) = L\mathbf{y} = \mathbf{b}$).

We first solve the linear system $L\mathbf{y} = \mathbf{b}$ by back-substitution from top to bottom since L is lower triangular. Writing out the equations, this system is

$$\begin{aligned} y_1 \qquad\qquad &= 1 \\ (1/5)y_1 + y_2 \qquad &= 3 \\ -(1/5)y_1 + 2y_2 + y_3 &= 4. \end{aligned}$$

The first equation tells us $y_1 = 1$, and plugging that into the second equation gives $(1/5) + y_2 = 3$, so $y_2 = 3 - 1/5 = 14/5$. Plugging these values for y_1 and y_2 into the third equation gives

$$-(1/5)1 + 2(14/5) + y_3 = 4,$$

which says $27/5 + y_3 = 4$, so $y_3 = 4 - 27/5 = 20/5 - 27/5 = -7/5$. Putting it all together gives

$$y_1 = 1, \quad y_2 = 14/5, \quad y_3 = -7/5.$$

Having found the vector $\mathbf{y}$, we now proceed to solve $U\mathbf{x} = \mathbf{y}$, which is an upper triangular system. Writing out the equations, this system is

$$\begin{aligned} 5x_1 + 0x_2 + \qquad x_3 &= 1 \\ 2x_2 + (9/5)x_3 &= 14/5 \\ -(7/5)x_3 &= -7/5 \end{aligned}$$

We will solve via back-substitution from bottom to top since it is upper triangular. The last equation tells us $x_3 = 1$, and plugging this into the next line up gives $2x_2 + (9/5)(1) = 14/5$, so $2x_2 = 1$ and hence $x_2 = 1/2$. Plugging these values for x_3 and x_2 into the first equation gives

$$5x_1 + 1 = 1,$$

so $x_1 = 0$.

Through these two steps we have found that the solution of the original equation is $\mathbf{x} = \begin{bmatrix} 0 \\ 1/2 \\ 1 \end{bmatrix}$. You can (and should) check that this really does satisfy $A\mathbf{x} = \mathbf{b}$.

To summarize, we solved $A\mathbf{x} = \mathbf{b}$ by successively solving two easier problems via back-substitution. Note that when solving for $\mathbf{y}$ in "$L\mathbf{y} = \mathbf{b}$" we went from top to bottom (i.e., y_1 to y_3) whereas when solving for $\mathbf{x}$ in "$U\mathbf{x} = \mathbf{y}$" we went from bottom to top (i.e., x_3 to x_1), ultimately because L is lower triangular and U is upper triangular. ■

We emphasize again that back-substitution works *only* if the diagonal entries of both L and U are all nonzero. The diagonal of L or U has a 0 whenever A is not invertible.

Suppose instead that we have factored a given *invertible* $n \times n$ matrix A as QR for $n \times n$ orthogonal Q and $n \times n$ upper triangular R with all diagonal entries nonzero. To solve $A\mathbf{x} = \mathbf{b}$, we use that $A\mathbf{x} = (QR)\mathbf{x} = Q(R\mathbf{x})$ to rewrite the system as $Q(R\mathbf{x}) = \mathbf{b}$ and so our task is two separate problems:

$$\text{solve } Q\mathbf{y} = \mathbf{b}, \quad \text{and then solve } R\mathbf{x} = \mathbf{y}.$$

In other words, we introduce an "intermediate" unknown n-vector $\mathbf{y}$ and solve the equation $Q\mathbf{y} = \mathbf{b}$ for this vector, and then use such $\mathbf{y}$ as the right side in the equation $R\mathbf{x} = \mathbf{y}$ (so $A\mathbf{x} = Q(R\mathbf{x}) = Q\mathbf{y} = \mathbf{b}$).

We have $Q^{-1} = Q^\top$ since Q is orthogonal, so the solution to $Q\mathbf{y} = \mathbf{b}$ is $\mathbf{y} = Q^\top\mathbf{b}$ (see Theorem 22.1.4). Now we know $\mathbf{y}$, and we can solve the resulting second equation $R\mathbf{x} = \mathbf{y}$ by back-substitution as in Example 22.1.1 since R is $n \times n$ upper triangular with all diagonal entries nonzero.

For the QR-decomposition of a general (perhaps non-invertible) $n \times n$ matrix, as well as a general $m \times n$ matrix, Q may be non-square (so not invertible) or R may have some 0's on its diagonal. Non-square Q arise in many applications; we address these possibilities in the optional Sections 22.4–22.5.

Example 22.2.4. Consider $A\mathbf{x} = \mathbf{b}$ given by

$$\begin{bmatrix} 1 & 3 & 2 \\ 1 & 1 & -2 \\ 1 & 2 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ 2 \end{bmatrix}.$$

Suppose we have been given the QR-decomposition of A:

$$\begin{bmatrix} 1 & 3 & 2 \\ 1 & 1 & -2 \\ 1 & 2 & -2 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{3} & 1/\sqrt{2} & 1/\sqrt{6} \\ 1/\sqrt{3} & -1/\sqrt{2} & 1/\sqrt{6} \\ 1/\sqrt{3} & 0 & -2/\sqrt{6} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 2\sqrt{3} & -2/\sqrt{3} \\ 0 & \sqrt{2} & 2\sqrt{2} \\ 0 & 0 & 4/\sqrt{6} \end{bmatrix}.$$

You can check by hand that the first matrix on the right side is an orthogonal matrix (its columns are an orthonormal set of vectors), and the second matrix on the right side is visibly upper triangular. This illustrates that even if the entries of A are simple numbers (e.g., integers), the entries of Q and R may be complicated-looking numbers. In particular, the presence of square roots is almost inevitable because each column of Q has length 1 (as Q is an orthogonal matrix). So although we don't yet know how such Q and R are found (it is discussed in Section 22.4), when given to us we can verify that they are correct.

Proceeding as we have learned, the solution $\mathbf{x}$ satisfies

$$R\mathbf{x} = Q^\top \mathbf{b} = Q^\top \begin{bmatrix} 4 \\ -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{6} & 1/\sqrt{6} & -2/\sqrt{6} \end{bmatrix} \begin{bmatrix} 4 \\ -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 5/\sqrt{3} \\ 5/\sqrt{2} \\ -1/\sqrt{6} \end{bmatrix},$$

and in terms of the explicit entries of R this is the triangular system

$$\begin{bmatrix} \sqrt{3} & 2\sqrt{3} & -2/\sqrt{3} \\ 0 & \sqrt{2} & 2\sqrt{2} \\ 0 & 0 & 4/\sqrt{6} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5/\sqrt{3} \\ 5/\sqrt{2} \\ -1/\sqrt{6} \end{bmatrix}.$$

We now solve this by back-substitution (going from bottom to top, since it is an upper triangular system). We first see that $(4/\sqrt{6})x_3 = -1/\sqrt{6}$, so $x_3 = -1/4$. The second equation says $\sqrt{2}\,x_2 + 2\sqrt{2}(-1/4) = 5/\sqrt{2}$, or $x_2 = 3$. Substituting the solved values of x_3 and x_2 into the top equation turns that into

$$\sqrt{3}\,x_1 + (2\sqrt{3})(3) + (-2/\sqrt{3})(-1/4) = 5/\sqrt{3},$$

or equivalently $\sqrt{3}\,x_1 + 6\sqrt{3} + (1/6)\sqrt{3} = (5/3)\sqrt{3}$, so $x_1 = -9/2$. We arrive, finally, at the unique solution $\mathbf{x} = \begin{bmatrix} -9/2 \\ 3 \\ -1/4 \end{bmatrix}$ (no ugly square roots!). You can (and should) check that this satisfies $A\mathbf{x} = \mathbf{b}$. ■

Example 22.2.5. For more practice with the QR-decomposition, let's solve $A\mathbf{x} = \mathbf{b}$ defined by

$$\begin{bmatrix} 2 & 1 & 1 \\ -1 & -2 & 1 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ -8 \\ 1 \end{bmatrix}$$

when given the QR-decomposition for A:

$$\begin{bmatrix} 2 & 1 & 1 \\ -1 & -2 & 1 \\ 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 2/\sqrt{6} & 0 & 1/\sqrt{3} \\ -1/\sqrt{6} & -1/\sqrt{2} & 1/\sqrt{3} \\ 1/\sqrt{6} & -1/\sqrt{2} & -1/\sqrt{3} \end{bmatrix} \begin{bmatrix} \sqrt{6} & 3/\sqrt{6} & 2/\sqrt{6} \\ 0 & 3/\sqrt{2} & -\sqrt{2} \\ 0 & 0 & 1/\sqrt{3} \end{bmatrix}.$$

As before, even though A has simple entries, the entries of Q and R are complicated.

Multiplying through by $Q^{\ \ 1} = Q^\top$ gives $R\mathbf{x} = Q^\top\mathbf{b}$, which in our case says

$$\begin{bmatrix} \sqrt{6} & 3/\sqrt{6} & 2/\sqrt{6} \\ 0 & 3/\sqrt{2} & -\sqrt{2} \\ 0 & 0 & 1/\sqrt{3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2/\sqrt{6} & -1/\sqrt{6} & 1/\sqrt{6} \\ 0 & -1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{3} & 1/\sqrt{3} & -1/\sqrt{3} \end{bmatrix} \begin{bmatrix} 7 \\ -8 \\ 1 \end{bmatrix} = \begin{bmatrix} 23/\sqrt{6} \\ 7/\sqrt{2} \\ -2/\sqrt{3} \end{bmatrix}.$$

Back-substitution (please check for yourself!) now yields $x_3 = -2$, then $x_2 = 1$, and finally $x_1 = 4$. You should also verify directly that $(x_1, x_2, x_3) = (4, 1, -2)$ satisfies the initial linear system. ■

Remark 22.2.6 (online resource). For any matrix, Wolfram quickly provides the QR- and LU-decompositions (as does matrixcalc). The matrix entries in the factors can be given as decimal approximations or in an exact form (using square roots, etc.). Playing around with software is a great way to get a feel for these decompositions: on the Wolfram website, search for the commands "QRDecomposition" and "LUDecomposition", and try some examples. (Capitalize the first three letters in these commands, since Wolfram is not smart enough to figure out your intent if you don't.)

Warning: There are several aspects of how Wolfram shows its answers that might cause confusion. The primary one is that the convention in Wolfram is to write a QR-decomposition in the form $Q^\dagger R$, where $(\cdot)^\dagger$ is Wolfram 's notation for "transpose" (Wolfram really says that this notation means "conjugate transpose", but that is a concept for matrices with complex numbers as entries and when it is applied to matrices with real number entries as in this book that means "transpose"). The "†" superscript is tiny on the screen, so it is easy to miss it. Since $(Q^\top)^\top = Q$, what WolframAlpha may seem to be calling "Q" if you don't look carefully enough is really $Q^\top$. If you plug Examples 22.2.4 and 22.2.5 into WolframAlpha's QRDecomposition command, what you'll get is correct but you have to be attentive to the tiny transpose superscript. The reason for this seemingly unusual convention is that $Q^\top$ is what gets used when solving a linear system, since $QR\mathbf{x} = \mathbf{b}$ with $Q^\top Q = \mathrm{I}_n$ (by orthogonality) implies $R\mathbf{x} = (Q^\top Q)R\mathbf{x} = Q^\top(QR\mathbf{x}) = Q^\top\mathbf{b}$ (with "$R\mathbf{x} = Q^\top\mathbf{b}$" an upper triangular linear system).

Other possible sources of confusion are discussed at the end of Examples 22.4.3, 22.4.6, and 22.4.8.

22.3. Computing inverses via matrix decompositions. Matrix decompositions are useful far beyond solving linear systems! We now use them to efficiently invert matrices; Section 22.5 (especially Proposition 22.5.1 onwards) and Appendix H discuss other important applications of QR-decompositions.

If A is an $n \times n$ matrix that has been written as a product of $n \times n$ matrices LU or QR *and* there are no 0's on the diagonal in L and U, or no 0's on the diagonal in R, then it is easy to compute A^{-1} as follows. The key point is that for *invertible* $n \times n$ matrices M and N, their product MN is invertible, with the inverse of their product equal to the product of inverses in the *reverse* order: $(MN)^{-1} = N^{-1}M^{-1}$. (This works by multiplying it out: $(N^{-1}M^{-1})(MN) = N^{-1}(M^{-1}M)N = N^{-1}\mathrm{I}_nN = N^{-1}N = \mathrm{I}_n$.) Hence,

$$\text{if } A = LU \text{ then } A^{-1} = U^{-1}L^{-1}, \quad \text{if } A = QR \text{ then } A^{-1} = R^{-1}Q^{-1} = R^{-1}Q^\top. \tag{22.3.1}$$

(We know that U^{-1}, L^{-1}, R^{-1} make sense when there are no 0's on the respective diagonals of U, L, R because upper triangular and lower triangular matrices are invertible exactly when they have no 0 on the diagonal, by Theorem 22.1.3.) This is useful because we'll soon see it is easy to compute the inverse of any upper and lower triangular matrix with no 0's on the diagonal.

Remark 22.3.1 (online resource). For matrix inverses, you can check answers with reshish, matrixcalc, or WolframAlpha.

Consider the upper triangular case (the lower triangular case goes similarly): suppose U (or R) has no 0's on the diagonal. Not only is U invertible, but in fact U^{-1} is *also* upper triangular. In effect, this is a consequence of back-substitution. The point is that for an upper triangular U, to solve $U\mathbf{x} = \mathbf{y}$ uniquely for $\mathbf{x}$ with any $\mathbf{y}$ is exactly the meaning of U being invertible (see Proposition 18.1.5) and is the same as saying $\mathbf{x} = U^{-1}\mathbf{y}$. Moreover, back-substitution gives a general formula for $\mathbf{x}$ in terms of $\mathbf{y}$, the coefficients of which are the entries of U^{-1}. As long as there is no 0 on the diagonal, the back-substitution method works perfectly (i.e., we really can always solve exactly for $\mathbf{x}$ in terms of $\mathbf{y}$ without running into inconsistencies), with x_i expressed in terms of $y_i, \dots, y_n$; this says U^{-1} is upper triangular.

In practice the back-substitution for computing A^{-1} for triangular $n \times n$ matrices A can be repackaged in another way. To explain this, let's first work out a 2×2 example.

Example 22.3.2. Let us compute $\begin{bmatrix} 2 & 4 \\ 0 & -6 \end{bmatrix}^{-1}$. One can use Example 18.2.1 to compute this inverse, but that method is specific to 2×2 matrices and we seek a different perspective that will adapt to upper triangular $n \times n$ matrices for any n.

We first make the "educated guess" that the inverse will also be upper triangular, with diagonal entries that are simply the reciprocals of the corresponding (nonzero) diagonal entries of U. That is, we guess that U^{-1} has the form of a matrix

$$U' = \begin{bmatrix} 1/2 & a \\ 0 & -1/6 \end{bmatrix}$$

for some number a that we seek.

Multiplying this guess by U (on the left or the right, it won't matter) yields

$$\begin{bmatrix} 1 & 2a - 4/6 \\ 0 & 1 \end{bmatrix}.$$

For this to equal I_2 (so our guess really is an inverse to U; see Theorem 18.1.8) says exactly that the upper-right entry $2a - 4/6$ is equal to 0, on in other words $a = 1/3$. To summarize, we have shown

$$\begin{bmatrix} 2 & 4 \\ 0 & -6 \end{bmatrix}^{-1} = \begin{bmatrix} 1/2 & 1/3 \\ 0 & -1/6 \end{bmatrix}.$$

■

This method will now be adapted to find the inverse U^{-1} for any $n \times n$ upper triangular matrix U with all diagonal entries nonzero. The inverse U^{-1} will always have 0's below the diagonal (this expresses that when solving $U\mathbf{x} = \mathbf{y}$ for $\mathbf{x}$ in terms of $\mathbf{y}$ by back-substitution, each x_i does not involve any y_j's for $j < i$), and its diagonal ii-entry will be the reciprocal $1/u_{ii}$ of the diagonal ii-entry u_{ii} of U (expressing the division step in back-substitution). We will compute the remaining unknown entries in U^{-1} above the diagonal one by one. To illustrate how this goes, let's work out a typical 3×3 case:

Example 22.3.3. Consider

$$U = \begin{bmatrix} 2 & 1 & 1 \\ 0 & -1 & 3 \\ 0 & 0 & 4 \end{bmatrix}, \quad \text{and guess that } U^{-1} \text{ has the form} \quad U' = \begin{bmatrix} 1/2 & a & b \\ 0 & -1 & c \\ 0 & 0 & 1/4 \end{bmatrix}$$

for some a, b, c that we seek. We compute

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \overset{?}{=} UU' = \begin{bmatrix} 2 & 1 & 1 \\ 0 & -1 & 3 \\ 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} 1/2 & a & b \\ 0 & -1 & c \\ 0 & 0 & 1/4 \end{bmatrix} = \begin{bmatrix} 1 & 2a - 1 & 2b + c + 1/4 \\ 0 & 1 & -c + 3/4 \\ 0 & 0 & 1 \end{bmatrix}.$$

The unknown entries a, b, c are now easy to find by setting all entries above the diagonal on the right side equal to 0 and working our way upwards to the upper-right corner: we first have the conditions

$$2a - 1 = 0, \quad -c + 3/4 = 0$$

by looking just above the diagonal, and this gives $a = 1/2$ and $c = 3/4$. Next, there is the vanishing condition from the upper-right corner:

$$2b + c + 1/4 = 0.$$

We already know the value for c, so plugging that into this equation gives $2b + 3/4 + 1/4 = 0$, which we solve to get $b = -1/2$. Hence,

$$U^{-1} = \begin{bmatrix} 1/2 & 1/2 & -1/2 \\ 0 & -1 & 3/4 \\ 0 & 0 & 1/4 \end{bmatrix}.$$

(Check for yourself that this answer is inverse to U.) ■

Completely analogous considerations can be used to find the inverses of lower triangular matrices (moving towards the lower left corner rather than towards the upper right corner as above), and the calculations are similar in the $n \times n$ case for any $n > 3$. Admittedly for $n > 3$ it begins to get a bit tedious, but hopefully you see that this is a fairly mechanical process, and the main issue when doing it by hand is not screwing up the arithmetic. *We will not ask you to work out such calculations with $n > 3$ on exams.*

The reason that computing the inverse for these matrices (when all diagonal entries are nonzero) is easier than computing the inverse of a general (invertible) $n \times n$ matrix is that the equations that we end up having to solve are quite simple. Indeed, what is really going on here is back-substitution in disguise. (In particular, the "guess" for the form of U^{-1} in Example 22.3.3 isn't really a guess since thinking in terms of back-substitution guarantees it will always have such a form. But you can think about it as a guess if you like and it will always work.)

Here is a 4×4 example, which we provide mainly to drive home the general pattern of working one's way up to the upper-right corner when solving the equations, moving up one "diagonal layer" at a time.

Example 22.3.4. Consider

$$U = \begin{bmatrix} -5 & -2 & 1 & -3 \\ 0 & 1 & 4 & -1 \\ 0 & 0 & 3 & 7 \\ 0 & 0 & 0 & -2 \end{bmatrix}, \quad \text{and guess that } U^{-1} \text{ has the form} \quad U' = \begin{bmatrix} -1/5 & a & b & c \\ 0 & 1 & d & e \\ 0 & 0 & 1/3 & f \\ 0 & 0 & 0 & -1/2 \end{bmatrix}$$

for some a, b, c, d, e, f that we seek. (As has been noted above, this isn't truly a "guess" since one can reason in general with back-substitution to be sure U^{-1} will have this form. But it is fine to think that we're just making an educated guess which one sees with experience always works.)

We compute

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \overset{?}{=} UU' = \begin{bmatrix} -5 & -2 & 1 & -3 \\ 0 & 1 & 4 & -1 \\ 0 & 0 & 3 & 7 \\ 0 & 0 & 0 & -2 \end{bmatrix} \begin{bmatrix} -1/5 & a & b & c \\ 0 & 1 & d & e \\ 0 & 0 & 1/3 & f \\ 0 & 0 & 0 & -1/2 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & -5a-2 & -5b-2d+1/3 & -5c-2e+f+3/2 \\ 0 & 1 & d+4/3 & e+4f+1/2 \\ 0 & 0 & 1 & 3f-7/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

To solve for the unknown entries a, b, c, d, e, f we set all entries above the diagonal to be 0, and it is simplest to systematically work our way to the upper-right corner by going along diagonal layers one at a time starting just above the actual diagonal. Namely, first looking at the entries just above the diagonal, we get the equations

$$-5a - 2 = 0, \quad d + 4/3 = 0, \quad 3f - 7/2 = 0.$$

These have nothing to do with each other (each equation involves its own variable not occurring in the other equations just above the diagonal), and so are solved to give $a = -2/5$, $d = -4/3$, $f = 7/6$. Note that these are exactly the entries in U' in the positions just above the diagonal.

Next, we go to the layer just on top of the one we solved, getting the equations

$$-5b - 2d + 1/3 = 0, \quad e + 4f + 1/2 = 0.$$

We already know d and f, so these are really just equations for b and e, again unrelated to each other. Plugging in the values for d and f, we solve to get $b = 3/5$ and $e = -31/6$. Note that b and f are also exactly the entries in U' in the positions corresponding to the layer we have just been working on.

Finally, just the upper-right corner remains to be used and correspondingly the only unknown left to be found is c, which is the upper-right entry of U'. Setting that entry to be 0 for UU' amounts to the equation

$$-5c - 2e + f + 3/2 = 0,$$

and plugging in the known values of e and f allows us to solve for c, obtaining $c = 13/5$. Putting it all together, we have arrived at

$$U^{-1} = \begin{bmatrix} -1/5 & -2/5 & 3/5 & 13/5 \\ 0 & 1 & -4/3 & -31/6 \\ 0 & 0 & 1/3 & 7/6 \\ 0 & 0 & 0 & -1/2 \end{bmatrix}.$$

Hopefully this 4×4 example conveys an even better sense than Example 22.3.3 for how the pattern of the calculation of U^{-1} goes with general upper triangular matrices whose diagonal entries are all nonzero. ■

22.4. QR-decomposition and Gram–Schmidt. We now explain how the QR-decomposition for a matrix A is just a repackaging of the Gram–Schmidt process for the columns of A; we will also give several worked numerical examples.

We shall primarily focus on the square $n \times n$ case; we also assume the columns of A are linearly independent, so A is invertible. You will only be asked to compute the QR-decomposition for invertible matrices; the resulting cases of the Gram–Schmidt process will be for a collection of vectors that is linearly independent, so at no step of the process will we obtain a zero vector.

To keep the notation simple, let us first focus on the case $n = 3$. Denote the columns of A by $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$; these are each vectors in $\mathbf{R}^3$. We assume that the $\mathbf{v}_i$'s are *linearly independent*, so when we apply the Gram–Schmidt process to them we obtain an orthogonal triple of nonzero vectors.

We shall make a small but important refinement in the Gram–Schmidt process that was absent in our discussion in Chapter 19: at the end we divide each of the (nonzero) "output" vectors $\mathbf{w}_i$ by its length to obtain a set of vectors that is orthonormal (not just mutually orthogonal). To avoid confusion with the notation used in our discussion of the Gram–Schmidt process in Chapter 19, we denote these new unit vectors as $\mathbf{w}'_1, \mathbf{w}'_2, \mathbf{w}'_3$ (so $\mathbf{w}'_i = \mathbf{w}_i/\|\mathbf{w}_i\|$).

Let us see how these unit vectors $\mathbf{w}'_i$ are related to the $\mathbf{v}_j$'s.

(i) In the first step, we have $\mathbf{w}_1 = \mathbf{v}_1$, so $\mathbf{w}'_1 = \mathbf{w}_1/\|\mathbf{w}_1\| = \mathbf{v}_1/\|\mathbf{v}_1\|$. In other words, $\mathbf{v}_1 = r_{11}\mathbf{w}'_1$ where r_{11} is the scalar $\|\mathbf{v}_1\| > 0$.

(ii) At the next step, the orthogonal pair of unit vectors $\mathbf{w}'_1, \mathbf{w}'_2$ has the same span as $\mathbf{w}_1, \mathbf{w}_2$, which has the same span as $\mathbf{v}_1, \mathbf{v}_2$, and by design $\mathbf{w}_2 = \mathbf{v}_2 - t\mathbf{w}_1$ for some scalar t. Hence, $\mathbf{v}_2 = t\mathbf{w}_1 + \mathbf{w}_2 = r_{12}\mathbf{w}'_1 + r_{22}\mathbf{w}'_2$ for some scalars r_{12} and r_{22}, with $r_{22} = \|\mathbf{w}_2\| > 0$.

(iii) Similarly, the orthonormal triple $\mathbf{w}'_1, \mathbf{w}'_2, \mathbf{w}'_3$ has the same span as $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, with $\mathbf{w}_3 = \mathbf{v}_3 - a\mathbf{v}_1 - b\mathbf{v}_2$ for some scalars a, b, so $\mathbf{v}_3 = a\mathbf{v}_1 + b\mathbf{v}_2 + \mathbf{w}_3 = r_{13}\mathbf{w}'_1 + r_{23}\mathbf{w}'_2 + r_{33}\mathbf{w}'_3$ for scalars r_{13}, r_{23}, r_{33} with $r_{33} = \|\mathbf{w}_3\| > 0$.

The Gram–Schmidt process gives formulas for the scalar coefficients $(r_{11}, r_{12}, r_{22}, r_{13}, r_{23}, r_{33})$ in terms of the dot products $\mathbf{v}_i \cdot \mathbf{v}_j$; we illustrate this explicitly in Example 22.4.3 below.

Now we write this in matrix language and see that it expresses exactly the QR-decomposition for A. Let Q be the 3×3 matrix whose columns are $\mathbf{w}'_1, \mathbf{w}'_2, \mathbf{w}'_3$ in turn; i.e.,

$$Q = \begin{bmatrix} | & | & | \\ \mathbf{w}'_1 & \mathbf{w}'_2 & \mathbf{w}'_3 \\ | & | & | \end{bmatrix}.$$

By design, the collection of columns of Q is orthonormal. The three equalities in (i), (ii), (iii) above correspond to the three equalities

$$Q\begin{bmatrix} r_{11} \\ 0 \\ 0 \end{bmatrix} = \mathbf{v}_1, \quad Q\begin{bmatrix} r_{12} \\ r_{22} \\ 0 \end{bmatrix} = \mathbf{v}_2, \quad Q\begin{bmatrix} r_{13} \\ r_{23} \\ r_{33} \end{bmatrix} = \mathbf{v}_3,$$

which when put together amounts to the matrix equation

$$Q\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix} = \begin{bmatrix} | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ | & | & | \end{bmatrix} = A.$$

But Q has *orthonormal* columns, so this is exactly the promised QR-decomposition of the matrix A (since the diagonal entries r_{11}, r_{22}, r_{33} of R are positive).

Moreover, we see that R also has a very concrete meaning in terms of the Gram–Schmidt process: its jth column consists of the coefficients $r_{1j}, \dots, r_{jj}$ that appear in the expression for $\mathbf{v}_j$ as a linear combination of $\mathbf{w}'_1, \dots, \mathbf{w}'_j$:

$$\mathbf{v}_j = r_{1j}\mathbf{w}'_1 + r_{2j}\mathbf{w}'_2 + \cdots + r_{jj}\mathbf{w}'_j$$

with $r_{jj} = \|\mathbf{w}_j\| > 0$ for each j. In practice those r_{ij}'s appear during the Gram–Schmidt process up to being multiplied by $\|\mathbf{w}_i\|$ once we account for the fact that $\mathbf{w}'_i = \mathbf{w}_i/\|\mathbf{w}_i\|$. This is illustrated in Example 22.4.2 and Example 22.4.3 below, so upon reading those you will see the general pattern explicitly.

Remark 22.4.1. If you read this entire discussion in reverse to go from matrix language back to statements about relations among column vectors, a QR-decomposition literally is the "Gram–Schmidt process" for $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$, up to the step of turning nonzero output vectors into unit vectors by dividing each by its length at the end. The main point is the Fourier formula (5.3.5) (or better: its derivation) that expresses the coefficients of a linear combination of orthogonal nonzero vectors in terms of dot products.

We now take a step back to see that these calculations actually accomplish a lot more than initially advertised: the same type of calculation yields a version of the QR-decomposition for general nonzero $m \times n$ matrices! To explain this, we review what the preceding calculations produce, as a series of three gradual generalizations;

(general m, independent columns) First, if you look back at this entire discussion, you can see that we *never* used that the columns $\mathbf{v}_i$ of A are vectors in $\mathbf{R}^3$. They could equally well be vectors in any $\mathbf{R}^m$. In other words, the calculations above work verbatim for any $m \times 3$ matrix A, so long as we still assume that the columns $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ of A are independent vectors.

Thus, any such matrix A can be written as a product $A = QR$ where Q is an $m \times 3$ matrix whose three columns are unit m-vectors $\mathbf{w}'_1, \mathbf{w}'_2, \mathbf{w}'_3$ that are mutually orthogonal, which is to say Q is an $m \times 3$ matrix with orthonormal columns, and R is a 3×3 upper triangular matrix with positive diagonal entries. This is the meaning of a QR-decomposition in this case.

(general $m \times n$, independent columns) Next, there is no reason to assume that there are 3 columns. We assumed this only to keep the notation under control. We can carry out the Gram–Schmidt process on any set of vectors $\mathbf{v}_1, \dots, \mathbf{v}_n$ in $\mathbf{R}^m$, and if we assume the $\mathbf{v}_i$'s are independent (so $n \le m$ since their span in $\mathbf{R}^m$ has dimension n) then we obtain an orthonormal set of vectors $\mathbf{w}'_1, \dots, \mathbf{w}'_n$. Define Q to be the $m \times n$ matrix whose columns are this new set of vectors $\mathbf{w}'_j$, so Q has orthonormal columns.

By construction, $\mathbf{v}_1 = r_{11}\mathbf{w}'_1$, $\mathbf{v}_2 = r_{12}\mathbf{w}'_1 + r_{22}\mathbf{w}'_2$, and in general $\mathbf{v}_j$ is a linear combination of $\mathbf{w}'_1, \dots, \mathbf{w}'_j$ using a positive coefficient r_{jj} for $\mathbf{w}'_j$, and the coefficients which appear in these linear combinations are the numbers that form the columns of an upper triangular $n \times n$ matrix R (so it has positive diagonal entries). The $m \times n$ matrix A with jth column $\mathbf{v}_j$ is $A = QR$. Non-square A with independent columns arise in *many* applications; see the optional Section 22.5 beginning at Example 22.5.3.

Let's work out two examples with square matrices, the only case arising on homework or exams.

Example 22.4.2. Consider the matrix

$$A = \begin{bmatrix} -2 & 1 \\ 1 & 3 \end{bmatrix}.$$

Denoting the columns of A from left to right as $\mathbf{v}_1$ and $\mathbf{v}_2$, by inspection neither is a scalar multiple of the other and so they are linearly independent. Gram–Schmidt for this pair is then essentially given by the formula in Theorem 7.1.1 (up to rearranging which vector comes first):

$$\mathbf{w}_1 = \mathbf{v}_1 = \begin{bmatrix} -2 \\ 1 \end{bmatrix}, \quad \mathbf{w}_2 = \mathbf{v}_2 - \mathbf{Proj}_{\mathbf{v}_1}(\mathbf{v}_2) = \mathbf{v}_2 - \frac{\mathbf{v}_2 \cdot \mathbf{v}_1}{\mathbf{v}_1 \cdot \mathbf{v}_1}\mathbf{v}_1 = \mathbf{v}_2 - (1/5)\mathbf{v}_1 = \begin{bmatrix} 7/5 \\ 14/5 \end{bmatrix} = \frac{7}{5}\begin{bmatrix} 1 \\ 2 \end{bmatrix},$$

so

$$\mathbf{v}_1 = \mathbf{w}_1, \quad \mathbf{v}_2 = (1/5)\mathbf{w}_1 + \mathbf{w}_2. \tag{22.4.1}$$

Since the unit vector $\mathbf{w}_i'$ is defined to be $\mathbf{w}_i/\|\mathbf{w}_i\|$, so $\mathbf{w}_i = \|\mathbf{w}_i\|\mathbf{w}_i'$, plugging this into (22.4.1) gives

$$\mathbf{v}_1 = \|\mathbf{w}_1\|\mathbf{w}_1', \quad \mathbf{v}_2 = (1/5)\|\mathbf{w}_1\|\mathbf{w}_1' + \|\mathbf{w}_2\|\mathbf{w}_2'. \tag{22.4.2}$$

Now comes the step where the square roots emerge: we compute the lengths of the $\mathbf{w}_i$'s from the explicit description of each of them as a 2-vector above:

$$\|\mathbf{w}_1\| = \sqrt{5}, \quad \|\mathbf{w}_2\| = \frac{7}{5}\sqrt{5} = \frac{7}{\sqrt{5}}. \tag{22.4.3}$$

This tells us R, by putting into the jth column the coefficients of $\mathbf{v}_j$ in (22.4.2) with the help of the explicit lengths in (22.4.3):

$$R = \begin{bmatrix} \sqrt{5} & (1/5)\sqrt{5} \\ 0 & 7/\sqrt{5} \end{bmatrix} = \begin{bmatrix} \sqrt{5} & 1/\sqrt{5} \\ 0 & 7/\sqrt{5} \end{bmatrix}.$$

(There is no need to pass to the final matrix on the right; we include it just to write the entries in a slightly "cleaner" form.)

Next we compute Q, whose columns are the vectors $\mathbf{w}_i' = \mathbf{w}_i/\|\mathbf{w}_i\|$. For this there is a handy trick to avoid some mess with fractions: for any nonzero vector $\mathbf{w}$ and *positive* scalar c we have $c\mathbf{w}/\|c\mathbf{w}\| = \mathbf{w}/\|\mathbf{w}\|$ (this just says that the unit vectors in the directions of $\mathbf{w}$ and $c\mathbf{w}$ are the same for any $c > 0$), so if we can factor out some *positive* scalar from some $\mathbf{w}_i$ to make it look cleaner then this has *no effect* on $\mathbf{w}_i' = \mathbf{w}_i/\|\mathbf{w}_i\|$. For instance, the above computation of $\mathbf{w}_2$ has the positive fraction $7/5$ factored out to leave behind something with integer entries, and to find $\mathbf{w}_2'$ we can just as well work with that integer vector (and its corresponding length).

Thus, we have

$$\mathbf{w}_1' = \frac{1}{\sqrt{5}}\begin{bmatrix} -2 \\ 1 \end{bmatrix} = \begin{bmatrix} -2/\sqrt{5} \\ 1/\sqrt{5} \end{bmatrix}, \quad \mathbf{w}_2' = \frac{1}{\sqrt{5}}\begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{5} \\ 2/\sqrt{5} \end{bmatrix}.$$

The entries in these two vectors give the columns of Q in order:

$$Q = \begin{bmatrix} -2/\sqrt{5} & 1/\sqrt{5} \\ 1/\sqrt{5} & 2/\sqrt{5} \end{bmatrix}.$$

As a safety check, you can verify by multiplying 2×2 matrices that QR indeed equals A. ■

Example 22.4.3. Consider the matrix

$$A = \begin{bmatrix} 1 & -1 & -2 \\ 0 & 1 & 1 \\ 2 & -1 & 2 \end{bmatrix}.$$

Denoting the columns of A from left to right as $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, we ran the Gram–Schmidt process on this triple in Example 19.3.6 to get

$$\mathbf{w}_1 = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}, \quad \mathbf{w}_2 = \begin{bmatrix} -2/5 \\ 1 \\ 1/5 \end{bmatrix} = \frac{1}{5}\begin{bmatrix} -2 \\ 5 \\ 1 \end{bmatrix}, \quad \mathbf{w}_3 = \begin{bmatrix} -5/3 \\ -5/6 \\ 5/6 \end{bmatrix} = \frac{5}{6}\begin{bmatrix} -2 \\ -1 \\ 1 \end{bmatrix}$$

and $\mathbf{w}_1 = \mathbf{v}_1, \mathbf{w}_2 = \mathbf{v}_2 + (3/5)\mathbf{w}_1, \mathbf{w}_3 = \mathbf{v}_3 - (2/5)\mathbf{w}_1 - (11/6)\mathbf{w}_2$, so

$$\mathbf{v}_1 = \mathbf{w}_1, \quad \mathbf{v}_2 = -(3/5)\mathbf{w}_1 + \mathbf{w}_2, \quad \mathbf{v}_3 = (2/5)\mathbf{w}_1 + (11/6)\mathbf{w}_2 + \mathbf{w}_3. \tag{22.4.4}$$

(One should always be careful when writing out the step corresponding to (22.4.4): it is very easy to make a sign error in the coefficients.)

Since the unit vector $\mathbf{w}_i'$ is defined to be $\mathbf{w}_i/\|\mathbf{w}_i\|$, so $\mathbf{w}_i = \|\mathbf{w}_i\|\mathbf{w}_i'$, plugging this into (22.4.4) gives

$$\mathbf{v}_1 = \|\mathbf{w}_1\|\mathbf{w}_1', \quad \mathbf{v}_2 = -\frac{3}{5}\|\mathbf{w}_1\|\mathbf{w}_1' + \|\mathbf{w}_2\|\mathbf{w}_2', \quad \mathbf{v}_3 = \frac{2}{5}\|\mathbf{w}_1\|\mathbf{w}_1' + \frac{11}{6}\|\mathbf{w}_2\|\mathbf{w}_2' + \|\mathbf{w}_3\|\mathbf{w}_3'. \tag{22.4.5}$$

Now comes the step where the square roots emerge: we compute the lengths of the $\mathbf{w}_i$'s from the explicit description of each of them as a 3-vector above:

$$\|\mathbf{w}_1\| = \sqrt{5}, \quad \|\mathbf{w}_2\| = \frac{1}{5}\sqrt{30} = \sqrt{\frac{6}{5}}, \quad \|\mathbf{w}_3\| = \frac{5}{6}\sqrt{6} = \frac{5}{\sqrt{6}}. \tag{22.4.6}$$

This tells us R, by putting into the jth column the coefficients of $\mathbf{v}_j$ in (22.4.5) with the help of the explicit lengths in (22.4.6):

$$R = \begin{bmatrix} \sqrt{5} & -(3/5)\sqrt{5} & (2/5)\sqrt{5} \\ 0 & \sqrt{6/5} & (11/6)\sqrt{6/5} \\ 0 & 0 & 5/\sqrt{6} \end{bmatrix} = \begin{bmatrix} \sqrt{5} & -3/\sqrt{5} & 2/\sqrt{5} \\ 0 & \sqrt{6/5} & 11/\sqrt{30} \\ 0 & 0 & 5/\sqrt{6} \end{bmatrix}.$$

(There is no need to pass to the final matrix on the right; we include it just to write the entries in a slightly "cleaner" form.)

Next we compute Q, whose columns are the vectors $\mathbf{w}_i' = \mathbf{w}_i/\|\mathbf{w}_i\|$. As in Example 22.4.2, for this there is a handy trick to avoid some mess with fractions: for any nonzero vector $\mathbf{w}$ and *positive* scalar c we have $c\mathbf{w}/\|c\mathbf{w}\| = \mathbf{w}/\|\mathbf{w}\|$ (this just says that the unit vectors in the directions of $\mathbf{w}$ and $c\mathbf{w}$ are the same for any $c > 0$), so if we can factor out some *positive* scalar from some $\mathbf{w}_i$ to make it look cleaner then this has *no effect* on $\mathbf{w}_i' = \mathbf{w}_i/\|\mathbf{w}_i\|$. For instance, the above expressions for $\mathbf{w}_2$ and $\mathbf{w}_3$ have positive fractions $1/5$ and $5/6$ factored out to leave behind something with integer entries, and to find $\mathbf{w}_i'$ we can just as well work with those integer vectors (and their corresponding lengths).

Thus, we have

$$\mathbf{w}_1' = \frac{1}{\sqrt{5}}\begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{5} \\ 0 \\ 2/\sqrt{5} \end{bmatrix}, \quad \mathbf{w}_2' = \frac{1}{\sqrt{30}}\begin{bmatrix} -2 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} -2/\sqrt{30} \\ 5/\sqrt{30} \\ 1/\sqrt{30} \end{bmatrix}, \quad \mathbf{w}_3' = \frac{1}{\sqrt{6}}\begin{bmatrix} -2 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -2/\sqrt{6} \\ -1/\sqrt{6} \\ 1/\sqrt{6} \end{bmatrix}.$$

The entries in these three vectors give the columns of Q in order:

$$Q = \begin{bmatrix} 1/\sqrt{5} & -2/\sqrt{30} & -2/\sqrt{6} \\ 0 & 5/\sqrt{30} & -1/\sqrt{6} \\ 2/\sqrt{5} & 1/\sqrt{30} & 1/\sqrt{6} \end{bmatrix}.$$

If we ask Wolfram for the QR-decomposition of this A (say as a safety check) then beware that it might write some matrix entries in a different-looking way but that doesn't mean our answer is wrong! For instance, when this example is entered into Wolfram sometimes the entry $11/\sqrt{30}$ in R is presented as $\sqrt{6/5} + \sqrt{5/6}$ (which is the same number, as you can convince yourself in any of a few ways: compare the decimal expressions to very many digits, directly manipulate square roots, or use that $\sqrt{a/b}+\sqrt{b/a}=(a+b)/\sqrt{ab}$ for any $a, b > 0$). The answers really are the same, even though WolframAlpha may write some matrix entries quite differently. ■

Remark 22.4.4. As a safety check, at the end of the preceding example you could multiply out QR to check that it is equal to A:

$$\begin{bmatrix} 1/\sqrt{5} & -2/\sqrt{30} & -2/\sqrt{6} \\ 0 & 5/\sqrt{30} & -1/\sqrt{6} \\ 2/\sqrt{5} & 1/\sqrt{30} & 1/\sqrt{6} \end{bmatrix} \begin{bmatrix} \sqrt{5} & -3/\sqrt{5} & 2/\sqrt{5} \\ 0 & \sqrt{6/5} & 11/\sqrt{30} \\ 0 & 0 & 5/\sqrt{6} \end{bmatrix} = \begin{bmatrix} 1 & -1 & -2 \\ 0 & 1 & 1 \\ 2 & -1 & 2 \end{bmatrix}.$$

Give this a try for yourself; it isn't too bad. In practice you'll always compute QR-decompositions on a computer, and software is always bug-free for such "standard" matrix procedures (or so everyone hopes!). But whatever algorithm is used, the main lesson of this section is that QR-decompositions are expressing exactly the output of the Gram–Schmidt process (after replacing each nonzero $\mathbf{w}_i$ with $\mathbf{w}_i/\|\mathbf{w}_i\|$).

Now we finally take up the remaining case for the QR-decomposition of a general nonzero matrix:

(general $m \times n$, dependent columns) What happens if the columns of A are linearly dependent? If we carry out the same calculations we have been doing then eventually we get a column $\mathbf{v}_j$ in the span of the previous columns $\mathbf{v}_1, \dots, \mathbf{v}_{j-1}$ (or the very first column $\mathbf{v}_1$ of A vanishes, a rare situation that is handled by putting an initial column of 0's in R at the end of the procedure). This means that we ignore $\mathbf{v}_j$ do not change the growing list $\mathbf{w}'_1, \dots, \mathbf{w}'_{j-1}$ that has been constructed up to this stage.

Since A has *some* nonzero column, at the end we have an orthonormal set $\mathbf{w}'_1, \dots, \mathbf{w}'_k$ in $\mathbf{R}^m$, where $k < n$ because we dropped at least one vector along the way. Define Q to be the $m \times k$ matrix whose columns are this orthonormal set, and by recording how each $\mathbf{v}_i$ is a linear combination of $\mathbf{w}'_1, \dots, \mathbf{w}'_k$, we obtain an upper triangular $k \times n$ matrix R so that $A = QR$. Some diagonal entry of R may vanish, such as when $k > 1$ but $\mathbf{v}_1$ and $\mathbf{v}_2$ are nonzero and scalar multiples of each other (then $r_{22} = 0$), but this doesn't always happen in cases with dependent columns (see Example 22.4.8).

Here is a worked example for the QR-decomposition in the non-invertible square case.

Example 22.4.5. Consider the matrix

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 2 \end{bmatrix}.$$

Denoting the columns of A from left to right as $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, running Gram–Schmidt on this triple yields

$$\mathbf{w}_1 = \mathbf{v}_1 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{w}_2 = \mathbf{v}_2 - \frac{1}{2}\mathbf{w}_1 = \begin{bmatrix} 1 \\ -1/2 \\ 1/2 \end{bmatrix}, \quad \mathbf{w}_3 = \mathbf{v}_3 - \frac{3}{2}\mathbf{w}_1 - \mathbf{w}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

(so the columns are linearly dependent, namely $\mathbf{v}_3 = (3/2)\mathbf{w}_1 + \mathbf{w}_2 = (3/2)\mathbf{v}_1 + (\mathbf{v}_2 - (1/2)\mathbf{v}_1) = \mathbf{v}_1 + \mathbf{v}_2$, so A is not invertible). Thus, the associated unit vectors $\mathbf{w}'_i = \mathbf{w}_i/\|\mathbf{w}_i\|$ are

$$\mathbf{w}'_1 = \begin{bmatrix} 0 \\ 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}, \quad \mathbf{w}'_2 = \begin{bmatrix} 2/\sqrt{6} \\ -1/\sqrt{6} \\ 1/\sqrt{6} \end{bmatrix}$$

and

$$\mathbf{v}_1 = \mathbf{w}_1 = \|\mathbf{w}_1\|\mathbf{w}'_1 = \sqrt{2}\mathbf{w}'_1, \quad \mathbf{v}_2 = \frac{1}{2}\mathbf{w}_1 + \mathbf{w}_2 = \frac{1}{2}\|\mathbf{w}_1\|\mathbf{w}'_1 + \|\mathbf{w}_2\|\mathbf{w}'_2 = \frac{1}{\sqrt{2}}\mathbf{w}'_1 + \sqrt{\frac{3}{2}}\mathbf{w}'_2,$$

$$\mathbf{v}_3 = \frac{3}{2}\mathbf{w}_1 + \mathbf{w}_2 = \frac{3}{2}\|\mathbf{w}_1\|\mathbf{w}_1' + \|\mathbf{w}_2\|\mathbf{w}_2' = \frac{3}{\sqrt{2}}\mathbf{w}_1' + \sqrt{\frac{3}{2}}\mathbf{w}_2'.$$

Hence,

$$A = \begin{bmatrix} 0 & 2/\sqrt{6} \\ 1/\sqrt{2} & -1/\sqrt{6} \\ 1/\sqrt{2} & 1/\sqrt{6} \end{bmatrix} \begin{bmatrix} \sqrt{2} & 1/\sqrt{2} & 3/\sqrt{2} \\ 0 & \sqrt{3/2} & \sqrt{3/2} \end{bmatrix} = QR$$

is the QR-decomposition of A. In this case Q is 3×2 with orthonormal columns and R is 2×3 upper triangular with positive diagonal entries, as promised in Remark 22.2.2 since $\dim C(A) = 2$ (indeed, $\dim C(A) < 3$ due to dependence of the columns, but the first two columns are linearly independent by inspection, so $\dim C(A) = 2$). ■

Here are two worked examples for non-square matrices, to provide you with some illustrations that should illuminate what we have been describing beyond the square case. (In Section 22.5 we discuss how to *use* the QR-decomposition beyond the invertible square case treated in Section 22.2.)

Example 22.4.6. Consider the matrix

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Denoting the columns of A from left to right as $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, we ran the Gram–Schmidt process on this triple in Example 19.3.1 to get

$$\mathbf{w}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{w}_2 = \begin{bmatrix} -1/2 \\ 1/2 \\ 1 \\ 0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -1 \\ 1 \\ 2 \\ 0 \end{bmatrix}, \quad \mathbf{w}_3 = \begin{bmatrix} 2/3 \\ -2/3 \\ 2/3 \\ 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 2 \\ -2 \\ 2 \\ 3 \end{bmatrix} \tag{22.4.7}$$

and $\mathbf{w}_1 = \mathbf{v}_1$, $\mathbf{w}_2 = \mathbf{v}_2 - (1/2)\mathbf{w}_1$, $\mathbf{w}_3 = \mathbf{v}_3 - (1/2)\mathbf{w}_1 - (1/3)\mathbf{w}_2$, so

$$\mathbf{v}_1 = \mathbf{w}_1, \quad \mathbf{v}_2 = (1/2)\mathbf{w}_1 + \mathbf{w}_2, \quad \mathbf{v}_3 = (1/2)\mathbf{w}_1 + (1/3)\mathbf{w}_2 + \mathbf{w}_3. \tag{22.4.8}$$

(At the risk of sounding repetitive, always be careful when writing out the step corresponding to (22.4.8): it is very easy to make a sign error in the coefficients.)

The unit vector $\mathbf{w}_i'$ is defined to be $\mathbf{w}_i/\|\mathbf{w}_i\|$, so $\mathbf{w}_i = \|\mathbf{w}_i\|\mathbf{w}_i'$. Plugging this into (22.4.8) gives

$$\mathbf{v}_1 = \|\mathbf{w}_1\|\mathbf{w}_1', \quad \mathbf{v}_2 = \frac{1}{2}\|\mathbf{w}_1\|\mathbf{w}_1' + \|\mathbf{w}_2\|\mathbf{w}_2', \quad \mathbf{v}_3 = \frac{1}{2}\|\mathbf{w}_1\|\mathbf{w}_1' + \frac{1}{3}\|\mathbf{w}_2\|\mathbf{w}_2' + \|\mathbf{w}_3\|\mathbf{w}_3'. \tag{22.4.9}$$

Now again comes the step where the square roots emerge: we compute the lengths of the $\mathbf{w}_i$'s from the explicit description of each of them as a 4-vector in (22.4.7) to obtain that

$$\|\mathbf{w}_1\| = \sqrt{2}, \quad \|\mathbf{w}_2\| = \sqrt{6}/2, \quad \|\mathbf{w}_3\| = \sqrt{21}/3. \tag{22.4.10}$$

This tells us R, by putting into the jth column the coefficients of $\mathbf{v}_j$ in (22.4.9) with the help of the explicit lengths in (22.4.10):

$$R = \begin{bmatrix} \sqrt{2} & (1/2)\sqrt{2} & (1/2)\sqrt{2} \\ 0 & \sqrt{6}/2 & \sqrt{6}/6 \\ 0 & 0 & \sqrt{21}/3 \end{bmatrix} = \begin{bmatrix} \sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \\ 0 & \sqrt{6}/2 & 1/\sqrt{6} \\ 0 & 0 & \sqrt{21}/3 \end{bmatrix}.$$

(Once again, there is no need to pass to the final matrix on the right; we include it just to write the entries in a slightly "cleaner" form.)

Next, we compute Q, whose columns are the vectors $\mathbf{w}'_i = \mathbf{w}_i/\|\mathbf{w}_i\|$. Using the same scaling trick as in the two preceding examples, we obtain

$$\mathbf{w}'_1 = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\\0\\0\end{bmatrix} = \begin{bmatrix}1/\sqrt{2}\\1/\sqrt{2}\\0\\0\end{bmatrix}, \quad \mathbf{w}'_2 = \frac{1}{\sqrt{6}}\begin{bmatrix}-1\\1\\2\\0\end{bmatrix} = \begin{bmatrix}-1/\sqrt{6}\\1/\sqrt{6}\\2/\sqrt{6}\\0\end{bmatrix}, \quad \mathbf{w}'_3 = \frac{1}{\sqrt{21}}\begin{bmatrix}2\\-2\\2\\3\end{bmatrix} = \begin{bmatrix}2/\sqrt{21}\\-2/\sqrt{21}\\2/\sqrt{21}\\3/\sqrt{21}\end{bmatrix}.$$

The entries in these three vectors give the columns of Q in order:

$$Q = \begin{bmatrix}1/\sqrt{2} & -1/\sqrt{6} & 2/\sqrt{21}\\ 1/\sqrt{2} & 1/\sqrt{6} & -2/\sqrt{21}\\ 0 & 2/\sqrt{6} & 2/\sqrt{21}\\ 0 & 0 & 3/\sqrt{21}\end{bmatrix}.$$

We conclude that the QR-decomposition of A says:

$$\begin{bmatrix}1/\sqrt{2} & -1/\sqrt{6} & 2/\sqrt{21}\\ 1/\sqrt{2} & 1/\sqrt{6} & -2/\sqrt{21}\\ 0 & 2/\sqrt{6} & 2/\sqrt{21}\\ 0 & 0 & 3/\sqrt{21}\end{bmatrix}\begin{bmatrix}\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2}\\ 0 & \sqrt{6}/2 & 1/\sqrt{6}\\ 0 & 0 & \sqrt{21}/3\end{bmatrix} = \begin{bmatrix}1 & 0 & 1\\ 1 & 1 & 0\\ 0 & 1 & 1\\ 0 & 0 & 1\end{bmatrix}.$$

If we ask Wolfram to compute the QR-decomposition of this A then it agrees with this but might not initially seem to agree since it may write some entries in a weird way. For instance, the entry $1/\sqrt{6}$ in R may sometimes appear as $\sqrt{2/3} - 1/\sqrt{6}$ (which really is the same number, as you can convince yourself by comparing decimal expressions or by directly manipulating square roots). ■

Remark 22.4.7. In the preceding example, the 4×3 matrix Q has columns that constitute an orthonormal set of three vectors in $\mathbf{R}^4$ (they're the vectors $\mathbf{w}'_1, \mathbf{w}'_2, \mathbf{w}'_3$ that are orthonormal by design). It is equivalent to say $Q^\top Q = \mathrm{I}_3$. However, the 4×4 matrix $QQ^\top$ given by multiplying Q and $Q^\top$ in the other order is very much unlike I_4 (as the rows of Q have no nice relations via dot products: Theorem 20.4.1 requires $m = n$ for good properties of rows). You can check by hand (or with a computer if you prefer) that

$$QQ^\top = \frac{1}{7}\begin{bmatrix}6 & 1 & -1 & 2\\ 1 & 6 & 1 & -2\\ -1 & 1 & 6 & 2\\ 2 & -2 & 2 & 3\end{bmatrix}.$$

Example 22.4.8. Consider the 4×3 matrix

$$A = \begin{bmatrix}2 & 0 & 2\\ 0 & 1 & 2\\ 1 & 1 & 3\\ 0 & 1 & 2\end{bmatrix}.$$

Denoting its columns in order as $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, we ran Gram–Schmidt on this triple in Example 19.3.2, where we got $\mathbf{w}_3 = \mathbf{0}$, from which a linear dependence relation among the $\mathbf{v}_i$'s was obtained: $\mathbf{v}_3 = \mathbf{v}_1 + 2\mathbf{v}_2$. So this matrix has linearly dependent columns. We will never ask you to work out a QR-decomposition in a case with dependent columns, and are providing such an example here just to illustrate the method.

The matrix Q is 4×2 (rather than 4×3) with columns $\mathbf{w}_1', \mathbf{w}_2'$ (due to the vanishing of $\mathbf{w}_3$). Following the general procedure for cases with dependent columns, we obtain

$$Q = \begin{bmatrix} 2/\sqrt{5} & -2/\sqrt{70} \\ 0 & 5/\sqrt{70} \\ 1/\sqrt{5} & 4/\sqrt{70} \\ 0 & 5/\sqrt{70} \end{bmatrix}, \quad R = \begin{bmatrix} \sqrt{5} & 1/\sqrt{5} & 7/\sqrt{5} \\ 0 & \sqrt{70}/5 & (2/5)\sqrt{70} \end{bmatrix}$$

and indeed $QR = A$. This Q has orthonormal columns, or equivalently $Q^\top Q = \mathrm{I}_2$ (beware that $QQ^\top$ is far from I_4, much like in Remark 22.4.7).

If you ask Wolfram, its Q and R agree with the ones above but – as we have mentioned in earlier examples – some matrix entries may look different yet really are the same number. For instance, the entry $(2/5)\sqrt{70}$ in R sometimes appears on Wolfram in the exotic-looking form $4\sqrt{2/35} + 2\sqrt{10/7}$ that really is the same number (as you can convince yourself by comparing decimal expressions or by directly manipulating with square roots). ■

If you want more practice computing QR-decompositions for A with linearly independent columns without first needing to do a Gram–Schmidt calculation from scratch, use Examples 19.3.7 and 19.3.8.

22.5. Using the QR-decomposition for general $m \times n$ matrices. We now want to describe how to use the QR-decomposition for general nonzero matrices, especially in the non-square case. (There is also a version of the LU-decomposition for general matrices, but it is notationally more complicated to describe and so we say nothing about it.) Let us first take a moment to review the various things we have done in this chapter related to linear systems.

We showed through examples in Section 22.1 that if A is upper-triangular (possibly non-square) with nonzero diagonal entries then it is easy to solve $A\mathbf{x} = \mathbf{b}$ and to determine the null space and column space for A. We then saw in Section 22.2 that if A is invertible and we know the decomposition $A = QR$ then we can use it to reduce solving $A\mathbf{x} = \mathbf{b}$ to two easier problems: solve $R\mathbf{x} = \mathbf{y}$ and solve $Q\mathbf{y} = \mathbf{b}$. The first of these two new problems is done via back-substitution (since R has nonzero diagonal entries, due to the invertibility of A), and the second can be solved by multiplying both sides of the equation by $Q^{-1} = Q^\top$. Finally, in Section 22.4 we saw how the two factors Q and R of this decomposition are determined by the data computed in the Gram–Schmidt process applied to the columns of A, formulated in a manner that makes sense for a general nonzero $m \times n$ matrix A.

The matrix Q is $m \times k$ with columns an orthonormal set of m-vectors, where k is the number of nonzero vectors obtained from the Gram–Schmidt process applied to the columns of A. In particular, $k = \dim C(A) \leq m$, so Q is a square matrix precisely when $C(A) = \mathbf{R}^m$ (which can fail). When Q is not square it has no inverse and so multiplying by Q^{-1} to solve a linear system does not make sense. However, as we now show, often we can still use the QR-decomposition of A to solve $A\mathbf{x} = \mathbf{b}$, as well as describe the null space and column space of A, and do many things with practical significance.

Let us revisit our general procedures for A a nonzero $m \times n$ matrix with arbitrary $m, n \geq 1$. Without making any assumption about linear independence of the columns of A, by applying the Gram–Schmidt process to the nonzero columns we arrive at a decomposition $A = QR$ where Q is an $m \times k$ matrix with orthonormal columns and R is an upper triangular $k \times n$ matrix for some $k \leq n$ (so R is of the type on the left in (22.1.2) if it is non-square); to define Q we have dropped any occurrence of the zero vector as an output in the Gram–Schmidt process, and the jth column of R vanishes when the jth column of A vanishes. How can we use this to solve the linear system $A\mathbf{x} = \mathbf{b}$?

As before, we break this into two problems, $Q\mathbf{y} = \mathbf{b}$ and $R\mathbf{x} = \mathbf{y}$. Thus, we should first determine $\mathbf{y}$ by solving $Q\mathbf{y} = \mathbf{b}$ and then use this intermediate solution $\mathbf{y}$ to solve $R\mathbf{x} = \mathbf{y}$. Since Q may not be square, there is no inverse and so it doesn't make sense to multiply by Q^{-1}. But the $k \times m$ matrix $Q^\top$ does make sense, and by Theorem 20.4.1 it is a "left inverse" to Q in the sense that the $k \times k$ matrix $Q^\top Q$ is equal to the $k \times k$ identity matrix I_k! (Beware that the $m \times m$ product $QQ^\top$ in the other order has nothing to do with an identity matrix in general since the *rows* of Q have no good relations via dot products when Q is not square; see Remark 22.4.7 for an explicit example.)

Now multiply both sides of $Q\mathbf{y} = \mathbf{b}$ by $Q^\top$ to get

$$Q^\top \mathbf{b} = Q^\top (Q\mathbf{y}) = (Q^\top Q)\mathbf{y} = \mathrm{I}_k \mathbf{y} = \mathbf{y}.$$

This works *regardless of the size and shape of A!* What is this new vector $Q^\top \mathbf{b}$? By definition of matrix-vector products, its components are the successive dot products of $\mathbf{b}$ with the rows of $Q^\top$, or equivalently with the columns $\mathbf{w}'_1, \ldots, \mathbf{w}'_k$ of Q. We have made a vector $\mathbf{y}$, and next want to find all solutions to $R\mathbf{x} = \mathbf{y}$ via back-substitution. It looks like everything is on track.

But there is a potential difficulty: we only showed that *if* $Q\mathbf{y} = \mathbf{b}$ has a solution then this solution must be $Q^\top \mathbf{b}$. We didn't check that this unique candidate for a solution actually works! A version of this issue arose in the discussion preceding Theorem 22.1.4, where we did succeed in verifying that a unique possible solution actually works in square cases. If we try the same here, we obtain $Q\mathbf{y} = Q(Q^\top \mathbf{b}) = (QQ^\top)\mathbf{b}$; this is equal to $\mathbf{b}$ (as desired) if $QQ^\top = \mathrm{I}_m$ but if Q is not square then $QQ^\top$ is nothing at all like an identity matrix (see Remark 22.4.7) even though $Q^\top Q = \mathrm{I}_k$.

So what is the meaning of $Q(Q^\top \mathbf{b})$ if it doesn't equal $\mathbf{b}$? We have already observed that the entries of the k-vector $Q^\top \mathbf{b}$ are the dot products $\mathbf{w}'_j \cdot \mathbf{b}$, so since Q has jth column $\mathbf{w}'_j$ it follows that

$$Q(Q^\top \mathbf{b}) = (\mathbf{w}'_1 \cdot \mathbf{b})\mathbf{w}'_1 + \ldots + (\mathbf{w}'_k \cdot \mathbf{b})\mathbf{w}'_k.$$

Since $\{\mathbf{w}'_1, \ldots, \mathbf{w}'_k\}$ is (by design) an orthonormal basis for $C(A)$ (in particular, the span $C(Q)$ of the vectors $\mathbf{w}'_j$ is equal to $C(A)$), the right side is the formula from (6.2.1) for projection of $\mathbf{b}$ into $C(A)$!

The upshot is that $\mathbf{y} = Q^\top \mathbf{b}$ is a solution to $Q\mathbf{y} = \mathbf{Proj}_{C(A)}(\mathbf{b})$ rather than to $Q\mathbf{y} = \mathbf{b}$, so solutions $\mathbf{x}$ to $R\mathbf{x} = \mathbf{y}$ actually satisfy $A\mathbf{x} = Q(R\mathbf{x}) = Q\mathbf{y} = \mathbf{Proj}_{C(A)}(\mathbf{b})$ rather than $A\mathbf{x} = \mathbf{b}$. In the special case $\mathbf{b} \in C(A)$, we are thereby getting exactly the desired solutions (but otherwise not). This applies to $\mathbf{b} = \mathbf{0} \in C(A)$, for which $\mathbf{y} = Q^\top \mathbf{b} = \mathbf{0}$, so solutions to $A\mathbf{x} = \mathbf{0}$ are the same as solutions to $R\mathbf{x} = \mathbf{0}$. Hence, $N(A) = N(R)$ always; we have just described $N(A)$ as the solutions to an upper triangular system $R\mathbf{x} = \mathbf{0}$ (which is easier to work with than the system $A\mathbf{x} = \mathbf{0}$).

The following result summarizes our conclusions so far (and this entire section is **optional**).

Proposition 22.5.1. Let A be a nonzero $m \times n$ matrix, and QR its QR-decomposition.

(i) The matrix Q is $m \times k$ with orthonormal columns and R is $k \times n$ upper triangular, where $k = \dim C(A) \leq m, n$.
(ii) If $k = m$ then Q is orthogonal with inverse $Q^\top$.
(iii) The column spaces $C(A)$ and $C(Q)$ are the same subspace of $\mathbf{R}^m$.
(iv) The null spaces $N(A)$ and $N(R)$ are the same subspace of $\mathbf{R}^n$, so $N(A)$ can be computed via back-substitution with R.
(v) For any $\mathbf{b} \in \mathbf{R}^m$, we have $QQ^\top \mathbf{b} = \mathbf{Proj}_{C(A)}\, \mathbf{b}$; i.e., $QQ^\top$ computes the projection into $C(A)$.

Remark 22.5.2. In many practical problems with data, one works with A for which k is very small compared to both m and n (e.g., m and n on the order of millions and $k < 100$). This occurs with recommender systems (such as for Netflix and Amazon), as we discussed in Section 21.6. Whenever we are in such a situation, the QR-decomposition is *tremendously useful* for efficiently computing products $A\mathbf{v}$ and AB against the $m \times n$ matrix A.

The point is that the associativity of matrix multiplication allows us to write such products as $(QR)\mathbf{v} = Q(R\mathbf{v})$ and $(QR)B = Q(RB)$, which are built up as a succession of left-multiplications by the $k \times n$ matrix R and then by the $m \times k$ matrix Q. Since k is *much smaller* than both m and n (i.e., R is "wide and thin", Q is "tall and thin"), such multiplications against Q and R involve *far fewer* arithmetic operations than multiplication against A, due to the smallness of k compared to both m and n.

We next push Proposition 22.5.1 further when the columns of A are **linearly independent** (so $k = n \leq m$). Here is a common practical context for this, with m much larger than n.

Example 22.5.3. In many scientific problems (in data science, electrical engineering, etc.), one needs to find a *small* number of constants in a mathematical model that optimally fit a *lot* of data; e.g., approximate an unknown function by a linear combination of a small number of known functions (the coefficients of the linear combination are the "constants" that one seeks). Here are two examples:

(i) There is noise in a communication channel, and we want to approximately reconstruct a signal over some time interval from measurements at specific times. If the unknown signal is a function $f : [0, T] \to \mathbf{R}$ on some time interval of length T and we have some known basic signals $f_1, \dots, f_8 : [0, T] \to \mathbf{R}$ (such as sine and cosine functions incorporating amplitude factors), then we seek coefficients $c_1, \dots, c_8$ so that the function f is well-approximated by the function $c_1 f_1 + \cdots + c_8 f_8$ across the *entire* time interval.

Making measurements of the signal at *many* times $t_1, \dots, t_m$ gives values $b_1, \dots, b_m$, and we hope that $f(t_i) \approx b_i$ for all i; typically m is much larger than the number $n = 8$ of basic signals. We seek $c_1, \dots, c_8$ for which the errors in the approximations (due to noise, etc.)

$$c_1 f_1(t_i) + c_2 f_2(t_i) + \cdots + c_8 f_8(t_i) \approx b_i \tag{22.5.1}$$

are collectively minimized. For the $m \times 8$ matrix $A = (f_j(t_i))$ of values of the known basic signals f_j at those times t_i, typically the 8 columns (corresponding to the 8 basic signals) are linearly independent. The left side of (22.5.1) is the ith entry of $A\mathbf{c} \in \mathbf{R}^m$ for the 8-vector $\mathbf{c}$ with entries $c_1, \dots, c_8$, so we want $A\mathbf{c} \approx \mathbf{b}$ in $\mathbf{R}^m$; in effect, $\mathbf{c}$ should be a "best approximate solution" to the massively overdetermined linear system $A\mathbf{x} = \mathbf{b}$ of m equations in 8 unknowns.

If the concept of "best approximate solution" can be made precise and then be computed, the resulting function $c_1 f_1(t) + \cdots + c_8 f_8(t)$ is our model for $f(t)$. Note that both A and $\mathbf{b}$ are built from the data (the times t_i, values $f_j(t_i)$, and noisy measurements b_i).

(ii) For the surface S of a body in a mechanical engineering problem, we want to approximate a function $f : S \to \mathbf{R}$ that expresses the values of a physical quantity of interest (e.g., heat, or force in some direction) when the body is subjected to specific experimental conditions.

For a small n there are some basic functions $f_1, \dots, f_n : S \to \mathbf{R}$ for which we hope to model f as a linear combination of the f_j's. At specific points $s_1, \dots, s_m$ on S, with m much larger than n, we make experimental measurements $b_1, \dots, b_m$ of f and will model f by the function $c_1 f_1 + \cdots + c_n f_n$ for coefficients $c_1, \dots, c_n$ that make the errors in the approximations

$$c_1 f_1(s_i) + \cdots + c_n f_n(s_i) \approx b_i$$

collectively minimized (in a sense we have to define).

As in (i), we will take $\mathbf{c} \in \mathbf{R}^n$ to be a "best approximate solution" (whatever that should mean) to the very overdetermined linear system $A\mathbf{x} = \mathbf{b}$ of m equations in n unknowns where $A =$

$(f_j(s_i))$ is the $m \times n$ matrix whose jth column records the values of the basic function f_j at the points $s_1, \dots, s_m \in S$. Typically the n columns of such A are linearly independent in $\mathbf{R}^m$.

The unified context for these and many other modeling problems is that we want to approximate an unknown function $f : R \to \mathbf{R}$ on some region R (a time interval, a surface in space, etc.) by a linear combination $c_1 f_1 + \cdots + c_n f_n$ of some known functions $f_1, \dots, f_n : R \to \mathbf{R}$ for a moderately small n, and we do this by taking the n-vector $\mathbf{c}$ with jth entry c_j to be the "best approximate solution" to $A\mathbf{x} = \mathbf{b}$ where:

- $A = (f_j(r_i))$ is the $m \times n$ matrix of values of the f_j's at many points $r_1, \dots, r_m \in R$,
- $\mathbf{b}$ is the m-vector whose ith entry b_i is the measured value of f at the point r_i in an experiment.

In practice the number m of sample measurements is much bigger than the number n of basic functions f_j, and the columns of A (which correspond to the f_j's) are linearly independent because the known basic functions f_j on R are sufficiently "unrelated" to each other and the columns record their values $f_j(r_i)$ at sufficiently many points $r_1, \dots, r_m$ in R. Zillions of data-fitting problems are special cases of this general task (after one strips away area-specific jargon to recognize the underlying mathematical problem), which we will soon see how to elegantly solve using the QR-decomposition of A. ■

Returning to the general setting of an $m \times n$ matrix A with linearly independent columns, we have $N(A) = \{\mathbf{0}\}$. (Indeed, the entries of any $\mathbf{v} \in N(A)$ are the coefficients of a linear relation among the columns of A, so the entries all vanish because the columns of A are linearly independent.) Thus, $A\mathbf{x}_1 = A\mathbf{x}_2$ forces $\mathbf{x}_1 = \mathbf{x}_2$ because $A(\mathbf{x}_1 - \mathbf{x}_2) = A\mathbf{x}_1 - A\mathbf{x}_2 = \mathbf{0}$ (implying $\mathbf{x}_1 - \mathbf{x}_2 \in N(A) = \{\mathbf{0}\}$). Hence, the linear system $A\mathbf{x} = \mathbf{b}$ has *at most one* solution (though it may not have a solution!).

The matrix Q from the QR-decomposition of A in Proposition 22.5.1 is $m \times n$ since $k = n$ (due to linear independence of the columns of A), and the matrix R is an upper triangular $n \times n$ matrix with all diagonal entries positive (due to the interpretation of R via Gram–Schmidt for the columns of A that we are assuming are linearly independent). The non-vanishing of the diagonal entries of R implies that R is invertible. Thus, $R^\top$ is also invertible.

We next observe that $A = QR$ implies $A^\top = R^\top Q^\top$, so

$$A^\top A = R^\top Q^\top Q R = R^\top (\mathrm{I}_k) R = R^\top R. \tag{22.5.2}$$

Since R and $R^\top$ are invertible, so is their product. Thus, even though A may not be invertible, the $n \times n$ matrix $A^\top A$ is *invertible* (keep in mind that we are assuming A has independent columns).

Multiplying both sides of "$A\mathbf{x} = \mathbf{b}$" on the left by $A^\top$ yields $(A^\top A)\mathbf{x} = A^\top \mathbf{b}$. This can be (uniquely) solved since $A^\top A$ is invertible, so multiplying both sides on the left by its inverse matrix motivates us to define

$$\mathbf{x}^* = (A^\top A)^{-1} A^\top \mathbf{b}. \tag{22.5.3}$$

The point of the preceding discussion is that *if* a solution actually exists for the original linear system $A\mathbf{x} = \mathbf{b}$ (where A again has independent columns) then this solution *must* be $\mathbf{x}^*$. We shall see that $\mathbf{x}^*$ is very useful even when it doesn't satisfy $A\mathbf{x} = \mathbf{b}$! To compute $\mathbf{x}^*$ efficiently, we use the QR-decomposition: since $A^\top A = R^\top R$ by (22.5.2), and $A^\top = (QR)^\top = R^\top Q^\top$, we have $\mathbf{x}^* = (R^\top R)^{-1}(R^\top Q^\top)\mathbf{b} = R^{-1}(R^\top)^{-1} R^\top Q^\top \mathbf{b} = R^{-1} Q^\top \mathbf{b}$. The expression $R^{-1} Q^\top$ here is much more efficient to use than $(A^\top A)^{-1} A^\top$ in the definition (22.5.3) when n is much smaller than m, which is to say A is "very tall" (as often happens); e.g., in Section 22.3 we saw how to efficiently compute inverses of triangular square matrices (such as R).

Just because a linear system has at most one candidate for a solution, it doesn't follow that this candidate must work (as we saw in the discussion preceding Proposition 22.5.1). Nonetheless, we have shown that the *only possible solution* to $A\mathbf{x} = \mathbf{b}$ (for A with independent columns) is $\mathbf{x}^*$ as defined above, and this works precisely when $\mathbf{b}$ is equal to

$$\begin{aligned} A\mathbf{x}^* = A(A^\top A)^{-1}A^\top\mathbf{b} = A(R^\top R)^{-1}A^\top\mathbf{b} &= (QR)(R^{-1}(R^\top)^{-1})(R^\top Q^\top)\mathbf{b} \\ &= Q(RR^{-1})((R^\top)^{-1}R^\top)Q^\top\mathbf{b} \\ &= Q(Q^\top\mathbf{b}) \\ &= \mathbf{Proj}_{C(A)}(\mathbf{b}) \end{aligned}$$

(the final equality uses Proposition 22.5.1(v)).

The equality of a vector with its projection into a subspace says that the vector lies in the subspace. Hence, $\mathbf{x}^*$ solves the original linear system precisely when $\mathbf{b} \in C(A)$. The distance-minimizing property of projections implies (still assuming A has independent columns) that $A\mathbf{x}^*$ is the point in $C(A)$ nearest to $\mathbf{b}$. Since $C(A)$ consists of exactly the points $A\mathbf{x}$ for varying $\mathbf{x}$, it follows that $\mathbf{x}^*$ makes $\|A\mathbf{x} - \mathbf{b}\|$ is as small as possible. Moreover, $\mathbf{x}^*$ is the *unique* n-vector for which $A\mathbf{x}^* = \mathbf{Proj}_{C(A)}(\mathbf{b})$ since we have seen that $A\mathbf{x}_1 = A\mathbf{x}_2$ implies $\mathbf{x}_1 = \mathbf{x}_2$. We have established:

Theorem 22.5.4 (Least squares (approximate) "solution"). For an $m \times n$ matrix A with independent columns (so $m \geq n$) and QR-decomposition QR, there is a unique $\mathbf{x}^* \in \mathbf{R}^n$ minimizing $\|A\mathbf{x} - \mathbf{b}\|$ and it is $\mathbf{x}^* = (A^\top A)^{-1}A^\top\mathbf{b} = R^{-1}Q^\top\mathbf{b}$. In particular, the linear system $A\mathbf{x} = \mathbf{b}$ has a solution precisely when $\mathbf{x}^*$ is a solution, which is to say $A(A^\top A)^{-1}A^\top\mathbf{b} = \mathbf{b}$ or equivalently $\mathbf{b}$ is in the null space of $A(A^\top A)^{-1}A^\top - \mathrm{I}_m = QQ^\top - \mathrm{I}_m$ (characterizing $\mathbf{b}$ for which $A\mathbf{x} = \mathbf{b}$ has a solution).

The vector $\mathbf{x}^*$ is called the *least squares* (*approximate*) *"solution"* to $A\mathbf{x} = \mathbf{b}$, even though it is usually not a solution. If a solution exists then it is the unique solution (recall that A has independent columns by hypothesis), and if one doesn't exist then it is the next best thing in a "least squares" sense: minimizing the distance from $A\mathbf{x}$ to $\mathbf{b}$. This give precise meaning to the idea of "best approximate solution" in the setting of Example 22.5.3, and it is efficiently computed via the QR-decomposition of A. In particular, for all situations as in Example 22.5.3, the matrix $(A^\top A)^{-1}A^\top = R^{-1}Q^\top$ applied to the vector $\mathbf{b}$ of sample measurements yields the best approximate "solution" (in the least squares sense). This matrix is a "left-inverse" to A in the sense that multiplying it on the left against A yields

$$((A^\top A)^{-1}A^\top)A = (A^\top A)^{-1}(A^\top A) = \mathrm{I}_n;$$

the matrix $(A^\top A)^{-1}A^\top$ is called the *pseudo-inverse* of such A (with independent columns!) and is often denoted A^+.

Example 22.5.5. Let's illustrate the preceding considerations with the 3×2 matrix

$$A = \begin{bmatrix} 2 & -1 \\ 3 & 4 \\ 1 & -2 \end{bmatrix}$$

and the linear system $A\mathbf{x} = \mathbf{b}$ for an arbitrary $\mathbf{b} \in \mathbf{R}^3$. The columns of A are independent (they are nonzero and not scalar multiples of each other), so Theorem 22.5.4 provides a least squares approximate "solution"

$$\mathbf{x}^* = (A^\top A)^{-1}A^\top\mathbf{b}$$

in terms of the entries b_1, b_2, b_3 of $\mathbf{b}$; this involves computing the inverse of the 2×2 matrix $A^\top A$. We are going to compute $\mathbf{x}^*$ in terms of the b_i's, and also give an explicit homogeneous system of 3 linear equations in the b_i's that characterizes when $A\mathbf{x} = \mathbf{b}$ actually has a solution (by making explicit the matrix $A(A^\top A)^{-1}A^\top - \mathrm{I}_3$ at the end of Theorem 22.5.4 whose null space consists of exactly those 3-vectors $\mathbf{b}$ for which $A\mathbf{x} = \mathbf{b}$ has a solution).

To calculate $(A^\top A)^{-1}A^\top$, let's work out the 2×2 matrix $A^\top A$ and invert it. We have

$$A^\top A = \begin{bmatrix} 2 & 3 & 1 \\ -1 & 4 & -2 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ 3 & 4 \\ 1 & -2 \end{bmatrix} = \begin{bmatrix} 14 & 8 \\ 8 & 21 \end{bmatrix}$$

whose determinant is $230 \neq 0$, so its inverse is $(A^\top A)^{-1} = \frac{1}{230}\begin{bmatrix} 21 & -8 \\ -8 & 14 \end{bmatrix}$. Hence, the least squares approximate "solution" is

$$\mathbf{x}^* = (A^\top A)^{-1}A^\top \mathbf{b} = \frac{1}{230}\begin{bmatrix} 21 & -8 \\ -8 & 14 \end{bmatrix}\begin{bmatrix} 2 & 3 & 1 \\ -1 & 4 & -2 \end{bmatrix}\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \frac{1}{230}\begin{bmatrix} 50 & 31 & 37 \\ -30 & 32 & -36 \end{bmatrix}\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$
$$= \frac{1}{230}\begin{bmatrix} 50b_1 + 31b_2 + 37b_3 \\ -30b_1 + 32b_2 - 36b_3 \end{bmatrix}.$$

Finally, $A\mathbf{x} = \mathbf{b}$ has a solution precisely when $\mathbf{b}$ is in the null space of the 3×3 matrix

$$A((A^\top A)^{-1}A^\top) - \mathrm{I}_3 = \begin{bmatrix} 2 & -1 \\ 3 & 4 \\ 1 & -2 \end{bmatrix}\left(\frac{1}{230}\begin{bmatrix} 50 & 31 & 37 \\ -30 & 32 & -36 \end{bmatrix}\right) - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \frac{1}{230}\left(\begin{bmatrix} 130 & 30 & 110 \\ 30 & 221 & -33 \\ 110 & -33 & 109 \end{bmatrix} - \begin{bmatrix} 230 & 0 & 0 \\ 0 & 230 & 0 \\ 0 & 0 & 230 \end{bmatrix}\right)$$
$$= \frac{1}{230}\begin{bmatrix} -100 & 30 & 110 \\ 30 & -9 & -33 \\ 110 & -33 & -121 \end{bmatrix}.$$

The null space of this is unaffected by ignoring the overall factor of $1/230$, so the linear system

$$\begin{aligned} -100b_1 + 30b_2 + 110b_3 &= 0 \\ 30b_1 - 9b_2 - 33b_3 &= 0 \\ 110b_1 - 33b_2 - 121b_3 &= 0 \end{aligned}$$

is satisfied by $\mathbf{b}$ precisely when $A\mathbf{x} = \mathbf{b}$ has a solution. This system of linear equations on $\mathbf{b} \in \mathbf{R}^3$ has $C(A)$ as its set of solutions, and it is a plane: all 3 equations in the system are scalar multiples of each other (look at ratios of coefficients of each b_i), and in fact of the equation $10b_1 - 3b_2 - 11b_3 = 0$. $\blacksquare$

Remark 22.5.6. For any $m \times n$ matrix A with independent columns (as above), among all $n \times m$ left-inverses M of A (i.e., $MA = \mathrm{I}_n$) the pseudo-inverse $A^+ = (A^\top A)^{-1}A^\top$ is the unique one with smallest "matrix norm" $\|M\| = \sqrt{\sum m_{ij}^2}$. Here is a sketch of a proof, treating $n \times m$ matrices as vectors in $\mathbf{R}^{nm}$ and using the concept of "matrix trace" $\mathrm{tr}(M) = \sum m_{ii}$.

For $n \times m$ matrices, define $B \cdot B' = \sum_{i,j} b_{ij}b'_{ij} = \sum_i(\sum_j b_{ij}b'_{ij}) = \sum_i(BB'^\top)_{ii} = \mathrm{tr}(BB'^\top)$; this is the usual dot product on $\mathbf{R}^{nm}$, and $B \cdot B = \sum b_{ij}^2 = \|B\|^2$, so the Pythagorean Theorem (in the sense of Theorem 2.3.1) applies for this "dot product" and the matrix norm. But $M - A^+$ and A^+ are orthogonal for this dot product (i.e., $(M - A^+) \cdot A^+ = 0$) because the matrix

$$(M - A^+)(A^+)^\top = MA(A^\top A)^{-1} - (A^\top A)^{-1}A^\top A(A^\top A)^{-1} = (MA - \mathrm{I}_n)(A^\top A)^{-1} = 0(A^\top A)^{-1}$$

vanishes and so has vanishing trace. Hence, the Pythagorean Theorem yields

$$\|M\|^2 = \|(M - A^+) + A^+\|^2 = \|M - A^+\|^2 + \|A^+\|^2 \geq \|A^+\|^2,$$

so $\|M\| \geq \|A^+\|$ with equality precisely when the excess term $\|M - A^+\|^2$ vanishes, which forces $M - A^+$ to vanish, or in other words $M = A^+$.

We emphasize that Theorem 22.5.4 is only applicable to linear systems $A\mathbf{x} = \mathbf{b}$ for which A has *linearly independent columns* (a condition sometimes referred to as "full rank"). If the columns of A are linearly dependent (so $N(A)$ contains nonzero vectors) then the $n \times n$ matrices R and $A^\top A$ in the preceding discussion are not invertible (since the null space of each contains the nonzero $N(A)$), so the procedures above Theorem 22.5.4 no longer make sense (there is no "R^{-1}" and no "$(A^\top A)^{-1}$"). Nonetheless, it always makes sense to seek an $\mathbf{x}$ minimizing $\|A\mathbf{x} - \mathbf{b}\|$ (and perhaps balancing minimization of $\|A\mathbf{x} - \mathbf{b}\|$ along with either $\|\mathbf{x}\|$ or lengths of other affine expressions in $\mathbf{x}$, a task called "multi-objective least squares" or "Tikhonov regularization" that arises in many economic, statistical, and scientific problems). This is a harder task when the columns are dependent, and it can be attacked via matrix factorizations and variants of the Lagrange multiplier method from Chapter 12.

Remark 22.5.7. The preceding technique of least squares approximate "solutions" to a linear system (with independent columns) computed using a QR-decomposition can be combined with the circle of ideas around the multivariable Newton's method from Section 18.5 to minimize the magnitude of *non-linear* vector-valued functions. The resulting technique, which in a basic form is due to Gauss, is called the Gauss–Newton method. The convergence issues for this can be quite delicate; it is one of many approaches to solving "non-linear least squares" problems.

Remark 22.5.8. In the preceding discussion we focused largely on overdetermined linear systems with independent columns. Sometimes one is confronted with the opposite scenario of an underdetermined linear system with *independent rows* (i.e., no "redundant equations"), so there is an abundance of solutions to $A\mathbf{x} = \mathbf{b}$ and then it is of interest to find an exact solution $\widetilde{\mathbf{x}}$ with minimal length. As a counterpart to Theorem 22.5.4, this new problem has a unique answer given moreover by a variant of the right side of (22.5.3) in terms of A and $A^\top$ (which as written makes no sense for underdetermined systems), and it can be computed via the QR-decomposition for $A^\top$ (rather than for A).

To explain this, note that the rows of A are the columns of $A^\top$, so Theorem 20.6.1 applies to $A^\top$ and thereby gives the invertibility of $(A^\top)^\top A^\top = AA^\top$. The matrix $A^\top(AA^\top)^{-1}$ therefore makes sense, and is a right-inverse of A in the sense that multiplying it against A on the right yields $A(A^\top(AA^\top)^{-1}) = (AA^\top)(AA^\top)^{-1} = \mathrm{I}_m$. The precise analogue of Theorem 22.5.4 is that the underdetermined linear system $A\mathbf{x} = \mathbf{b}$ has a unique exact solution with minimal length, given by

$$\widetilde{\mathbf{x}} = A^\top(AA^\top)^{-1}\mathbf{b}.$$

To prove this assertion, first observe that this vector does indeed satisfy $A\mathbf{x} = \mathbf{b}$ (check, using that $A^\top(AA^\top)^{-1}$ is a right-inverse to A). To see that it is the unique solution with minimal length, we shall use the Pythagorean Theorem (i.e., Theorem 2.3.1) and the interaction of transpose with dot products in (20.1.2). For any n-vector $\mathbf{x}'$ satisfying $A\mathbf{x}' = \mathbf{b}$, the difference vector $\mathbf{x}' - \widetilde{\mathbf{x}}$ is orthogonal to $\widetilde{\mathbf{x}}$:

$$\begin{aligned}(\mathbf{x}' - \widetilde{\mathbf{x}}) \cdot \widetilde{\mathbf{x}} = (\mathbf{x}' - \widetilde{\mathbf{x}}) \cdot A^\top(AA^\top)^{-1}\mathbf{b} = A(\mathbf{x}' - \widetilde{\mathbf{x}}) \cdot (AA^\top)^{-1}\mathbf{b} &= (A\mathbf{x}' - A\widetilde{\mathbf{x}}) \cdot (AA^\top)^{-1}\mathbf{b} \\ &= (\mathbf{b} - \mathbf{b}) \cdot (AA^\top)^{-1}\mathbf{b} \\ &= \mathbf{0} \cdot (AA^\top)^{-1}\mathbf{b} \\ &= 0\end{aligned}$$

(the second equality uses (20.1.2), the fourth uses that $A\mathbf{x}' = \mathbf{b}$). Thus, by the Pythagorean Theorem,

$$\|\mathbf{x}'\|^2 = \|(\mathbf{x}' - \widetilde{\mathbf{x}}) + \widetilde{\mathbf{x}}\|^2 = \|\mathbf{x}' - \widetilde{\mathbf{x}}\|^2 + \|\widetilde{\mathbf{x}}\|^2 \geq \|\widetilde{\mathbf{x}}\|^2$$

with equality precisely when the excess term $\|\mathbf{x}' - \widetilde{\mathbf{x}}\|^2$ vanishes, which forces $\mathbf{x}' - \widetilde{\mathbf{x}} = \mathbf{0}$, or equivalently $\mathbf{x}' = \widetilde{\mathbf{x}}$. Taking square roots in the inequality gives that $\|\mathbf{x}'\| \geq \|\widetilde{\mathbf{x}}\|$ with equality precisely when $\mathbf{x}' = \widetilde{\mathbf{x}}$, so $\widetilde{\mathbf{x}}$ is the unique length minimizer among solutions to $A\mathbf{x} = \mathbf{b}$.

Computing the length-minimizing solution $\widetilde{\mathbf{x}}$ efficiently can be done using a QR-decomposition for the transpose matrix $A^\top$ (rather than for A!), as we now explain. The independence of the rows of A is the same as the independence of the *columns* of $A^\top$, so in the QR-decomposition $A^\top = Q'R'$ the matrix R' is invertible. Orthonormality of the columns of the non-square Q' gives that $Q'^\top Q' = \mathrm{I}_m$, so

$$AA^\top = (A^\top)^\top A^\top = (Q'R')^\top Q'R' = R'^\top Q'^\top Q'R' = R'^\top (Q'^\top Q')R' = R'^\top R'$$

analogous to (22.5.2). Hence

$$A^\top (AA^\top)^{-1} = (Q'R')(R'^\top R')^{-1} = Q'R'R'^{-1}(R'^\top)^{-1} = Q'(R'^\top)^{-1},$$

so $\widetilde{\mathbf{x}} = Q'(R'^\top)^{-1}\mathbf{b}$; this is an explicit and efficient formula for the unique minimal-length solution of an underdetermined linear system $A\mathbf{x} = \mathbf{b}$ for which A has independent rows (and we again emphasize that $Q'R'$ is the QR-decomposition of $A^\top$, not of A).

22.6. Finding an LU-decomposition using "row reduction". In this section we describe how to find an LU-decomposition for an $n \times n$ matrix A. The traditional method, as explained below, amounts to a technique called "row reduction" or "Gaussian elimination". This adapts to any $m \times n$ matrix, but the notation gets cumbersome, so we focus on the $n \times n$ case here. If you have studied some linear algebra prior to this course then you may have encountered row reduction before; that is the main reason we are explaining in this section how the LU-decomposition is a reinterpretation of row reduction. But we should emphasize at the outset that row reduction as an algorithm is obsolete; when modern computers find LU-decompositions, they generally use other methods that are faster.

If we were to carry out Example 22.1.1 in full detail, we would see that we were applying a certain sequence of operations to solve the system. In general the collection of *solutions* to $A\mathbf{x} = \mathbf{b}$ are unchanged if we carry out a sequence of *basic row operations* which replace the matrix A, and at the same time the vector $\mathbf{b}$ on the right of the equation, by a sequence of new matrices and vectors obtained by

- adding a multiple of one row to another row;
- multiplying a given row by a nonzero scalar,
- interchanging any two rows.

This is the same as carrying out these operations on the individual linear equations which make up the system: we can add a multiple of one equation to another of the equations, we can multiply any one of the equations by a nonzero scalar, or we can interchange the order in which we list the equations. What this has to do with matrix decompositions is that we can encode this sequence of operations by matrix multiplication. Note that the actual equations in the system are changing, but the collection of solution vectors (which is what we care about) does *not* change.

We illustrate this idea with a specific example. Let $A = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 5 & 1 \\ 4 & 2 & 3 \end{bmatrix}$, and suppose that we wish to solve the equation $A\mathbf{x} = \mathbf{b}$ via the LU-decomposition method. We do not specify $\mathbf{b}$ here since we want to show that the manipulations only depend on A. In the following, we write the rows of A as R_1, R_2, R_3 (these are row vectors).

At the level of the system of equations, the operations we would like to do include adding a multiple of one equation to another lower down in the list and multiplying an equation by a nonzero scalar. You might want to add a multiple of one equation to another *higher up* in the list of equations, or swap two of the equations. In the general process called row reduction (or Gaussian elimination), these extra two operations are allowed. We only wish to include the first two operations (for reasons to be explained). That is, at the level of working with the matrix, the two operations we allow are: adding a multiple of one row R_i to R_j *when* $i < j$, and multiplying any row R_i by a nonzero scalar.

Let us now do a sequence of such operations on this particular matrix:

$$\underbrace{\begin{bmatrix} 1 & 0 & -1 \\ 2 & 5 & 1 \\ 4 & 2 & 3 \end{bmatrix}}_{A} \xrightarrow{R_3 \to R_3 - 4R_1} \underbrace{\begin{bmatrix} 1 & 0 & -1 \\ 2 & 5 & 1 \\ 0 & 2 & 7 \end{bmatrix}}_{A_1} \xrightarrow{R_2 \to R_2 - 2R_1} \underbrace{\begin{bmatrix} 1 & 0 & -1 \\ 0 & 5 & 3 \\ 0 & 2 & 7 \end{bmatrix}}_{A_2} \xrightarrow{R_3 \to R_3 - 2R_2/5} \underbrace{\begin{bmatrix} 1 & 0 & -1 \\ 0 & 5 & 3 \\ 0 & 0 & 29/5 \end{bmatrix}}_{U}$$

We keep on doing such operations until we arrive at an upper triangular matrix U. There may be many ways in which way we do the operations, and we might have chosen a completely different set of specific operations (and maybe even arrived at a different upper triangular matrix).

Remark 22.6.1. There is no guarantee that this method will be successful. For example, suppose at some stage we arrive at a matrix

$$\begin{bmatrix} * & 0 & * \\ 0 & 0 & * \\ * & * & * \end{bmatrix}$$

where each of the $*$'s are nonzero. There is no way to add multiples of the first and second rows to the third row to obtain an upper triangular matrix. This is a case where an LU-decomposition really does not exist. Such difficulties correspond to the presence of 0 in specific positions during the application of row operations, and (in a sense we won't make precise) this is a "rare" phenomenon.

In our particular example, we found a good sequence of operations and arrived at some U. Now let us interpret these row operations. The first operation, where $-4R_1$ is added to R_3, can be interpreted by multiplying A on the left by a lower triangular matrix:

$$A_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix} A;$$

Continuing on with the row operations,

$$A_2 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} A_1$$

and

$$U = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2/5 & 1 \end{bmatrix} A_2.$$

Putting these all together, what we have done amounts to multiplying A on the left by three lower triangular matrices:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2/5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix} A = U.$$

The order matters a lot – we cannot interchange the order of these matrix multiplications!

The next step is to note that it is very easy to find the inverse of any one of these "elementary" lower triangular matrices. For example,

$$\begin{bmatrix}1&0&0\\0&1&0\\-4&0&1\end{bmatrix}^{1} = \begin{bmatrix}1&0&0\\0&1&0\\4&0&1\end{bmatrix}$$

We simply change the sign on the single nonzero entry below the diagonal. This means that we can move each lower triangular term to the right side by multiplying by its inverse; i.e.,

$$A = \begin{bmatrix}1&0&0\\0&1&0\\4&0&1\end{bmatrix}\begin{bmatrix}1&0&0\\2&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}1&0&0\\0&1&0\\0&2/5&1\end{bmatrix}U.$$

The final step is to observe that the product of any two – and hence any number of – lower triangular matrices is lower triangular! (Try a couple of 3×3 examples for yourself to see why this works. The analogue holds for the upper triangular case.) Thus, if we define L to be the product of the three lower triangular matrices here, it is still lower triangular. Carrying this out gives

$$L = \begin{bmatrix}1&0&0\\2&1&0\\4&2/5&1\end{bmatrix}.$$

We have now produced our LU-decomposition of A:

$$\begin{bmatrix}1&0&-1\\2&5&1\\4&2&3\end{bmatrix} = \begin{bmatrix}1&0&0\\2&1&0\\4&2/5&1\end{bmatrix}\begin{bmatrix}1&0&-1\\0&5&3\\0&0&29/5\end{bmatrix}.$$

The same principle applies if we are working with a matrix of any size. Namely, we carry out a sequence of the two types of allowable row operations: add a multiple of R_i to R_j *only for* $i < j$, and multiply R_i by a scalar $c \neq 0$. Each of these corresponds to multiplying on the left by two kinds of "elementary" lower triangular matrices: a matrix with 1's on the diagonal and exactly one nonzero entry below the diagonal (in the ji-entry), or a matrix with a nonzero scalar c in the diagonal ii-entry, 1's elsewhere on the diagonal, and 0's everywhere off the diagonal. For "most" A we can find a sequence of such operations and arrive at an upper triangular matrix:

$$L_N L_{N\;\,1} \dots L_2 L_1 A = U.$$

We then successively multiply by the inverses of these lower triangular matrices to move all of the L_j to the other side:

$$L_{N\;\,1} \dots L_1 A = L_N^{1} U, \quad L_{N\;\,2} \dots L_1 A = L_{N\;\,1}^{1} L_N^{1} U, \quad \dots, \quad A = L_1^{1} \dots L_N^{1} U,$$

and hence $A = LU$ for $L = L_1^{1} \dots L_N^{1}$.

As we observed along the way, there was no absolutely natural way to decide on which row operations to do, and in which order, which means that we might have chosen different factors L_j along the way. This means that the two factors L and U here are not uniquely determined by A alone (they may depend on choices we made along the way). There is a refinement of the LU-decomposition called the LDU-decomposition that avoids such failure of uniqueness.

Remark 22.6.2 (*LU* with partial pivots). We pointed out in Remark 22.6.1 that it is sometimes impossible to carry out a sequence of the two basic row operations used above to obtain an upper triangular matrix (and that an LU-decomposition can sometimes fail to exist). The cause of the difficulty is that we did not allow one further type of row operation: interchanging row i with row j. The reason we avoided this is that the swapping of R_i and R_j corresponds to multiplying on the left by a special type of matrix which is **not** lower triangular! For example, the swapping of the two rows of a 2×2 matrix is expressed as

$$A = \begin{bmatrix} 2 & 3 \\ 4 & -1 \end{bmatrix} \rightsquigarrow \begin{bmatrix} 4 & -1 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 4 & -1 \end{bmatrix} = PA$$

where left multiplication by the 2×2 matrix P encodes swapping the two rows.

So we *can* transform a matrix to become upper triangular by a broader sequence of row operations, and hence via a sequence of left multiplications by certain "elementary" matrices, but not always using just left multiplications by lower triangular matrices. With some more care, it can be shown that the need to deviate from lower triangular left multiplication can be avoided by applying a suitable rearrangement (also called "permutation") of the rows *at the start*. This yields a final decomposition $A = PLU$ with upper triangular U, lower triangular L, and a "permutation matrix" P (a matrix having a single entry of 1 in each row and in each column, with all other entries equal to 0); the process is called "LU with partial pivots".

Remark 22.6.3. To conclude our discussion of upper and lower triangular matrices, let's address a point that has been mentioned several times in this chapter: an upper or lower triangular $n \times n$ matrix is never invertible when it has some diagonal entry equal to 0. Our experience with back-substitution explains why invertibility holds when the diagonal entries are all nonzero, and now let's see why invertibility really cannot ever hold if some diagonal entry does vanish. We discuss this in the upper triangular case; the lower triangular case follows by applying transposes.

Suppose U is an upper triangular matrix, and that some diagonal entry u_{ii} vanishes. We will show that $N(U)$ is nonzero, so U cannot be invertible. Since $\dim N(U) + \dim C(U) = n$ by the Rank-Nullity Theorem, it then also follows that the linear subspace $C(U)$ in $\mathbf{R}^n$ has dimension $< n$.

If $i = 1$ then the first column of U vanishes (since U is upper triangular with $u_{11} = 0$), so $U\mathbf{e}_1 = \mathbf{0}$ and hence $\mathbf{e}_1 \in N(U)$ (so $N(U)$ is nonzero). Suppose instead that $i > 1$, and look at the effect of U on $\mathbf{e}_1, \dots, \mathbf{e}_i$. Since U is upper triangular, the jth column $U\mathbf{e}_j$ of U belongs to the span of $\mathbf{e}_1, \dots, \mathbf{e}_j$. In particular, for $j \leq i - 1$ we have

$$U\mathbf{e}_j \in \mathrm{span}(\mathbf{e}_1, \dots, \mathbf{e}_j) \subset \mathrm{span}(\mathbf{e}_1, \dots, \mathbf{e}_{i-1}).$$

The vanishing of u_{ii} says that $U\mathbf{e}_i$ belongs to the span of $\mathbf{e}_1, \dots, \mathbf{e}_{i-1}$ too, so the i vectors $U\mathbf{e}_1, \dots, U\mathbf{e}_i$ all lie in the $(i-1)$-dimensional subspace $\mathrm{span}(\mathbf{e}_1, \dots, \mathbf{e}_{i-1})$.

We have just built i vectors in a subspace of dimension $i - 1$. That is more vectors than the dimension of this subspace, so it cannot be a linearly independent collection (or else its span would have dimension i, which is impossible inside a subspace of dimension $i - 1$). Thus, this collection of vectors is linearly dependent, so there exist scalars $c_1, \dots, c_i$ not all equal to 0 for which

$$c_1 U\mathbf{e}_1 + \cdots + c_i U\mathbf{e}_i = \mathbf{0},$$

or equivalently

$$U(c_1\mathbf{e}_1 + \cdots + c_i\mathbf{e}_i) = \mathbf{0}.$$

This says that the n-vector $c_1\mathbf{e}_1 + \cdots + c_i\mathbf{e}_i$ belongs to $N(U)$, and it is nonzero since some c_j is nonzero (remember the meaning of $\mathbf{e}_1, \dots, \mathbf{e}_i$). We have shown that $N(U)$ is nonzero, as desired.

Chapter 22 highlights (links to highlights in and next chapters)

Notation	**Meaning**	**Location in text**
nothing new!		

Concept	**Meaning**	**Location in text**
back-substitution	technique to solve a triangular system of linear equations by working from each equation to the next beginning with the one having the fewest variables	Example 22.1.1
LU-decomposition (for square case)	expressing a square matrix as a product of lower and upper triangular square matrices	Theorem 22.2.1(i)
QR-decomposition (for invertible case)	expressing an invertible matrix A as a product of an orthogonal matrix (Q) and an upper triangular matrix (R) with positive diagonal entries	Theorem 22.2.1(ii)

Result	**Meaning**	**Location in text**
upper or lower triangular with nonzero diagonal entries is invertible	if A is $n \times n$ triangular (upper or lower) and all $a_{ii} \neq 0$ then A is invertible	Theorem 22.1.3
for orthogonal $n \times n$ matrix A, $A\mathbf{x} = \mathbf{b}$ has exactly one solution	$A^{\top}\mathbf{b}$ is the unique solution	Theorem 22.1.4
triangular systems with nonzero diagonal coefficients can be analyzed via back-substitution	for $m \times n$ matrix A that is upper or lower triangular, back-substitution completely describes solutions to $A\mathbf{x} = \mathbf{b}$: unique if $m = n$, with $n - m$ parameters if $m < n$, and $m - n$ equations on b_i's if $m > n$	Exs. 22.1.5-22.1.8
existence of LU-decomposition and QR-decomposition in square cases	for $n \times n$ matrices A, most have LU-decomposition and invertible ones have QR-decomposition	Theorem 22.2.1
for invertible matrix A, can compute A^{-1} from LU-decomposition and from QR-decomposition	$(LU)^{-1} = U^{-1}L^{-1}$ & $(QR)^{-1} = R^{-1}Q^{\top}$, with each of U^{-1}, L^{-1}, R^{-1} computable via back-substitution	(22.3.1)

Skill	**Location in text**
use back-substitution to solve $A\mathbf{x} = \mathbf{b}$ for upper or lower triangular A with nonzero diagonal entries (parameters in underdetermined case, equations on $\mathbf{b}$-entries in overdetermined case)	Ex. 22.1.1, Exs. 22.1.5-22.1.8, Theorem 22.1.9
use LU-decomposition to solve $A\mathbf{x} = \mathbf{b}$ for square A	Example 22.2.3
use QR-decomposition to solve $A\mathbf{x} = \mathbf{b}$ for invertible A	Examples 22.2.4, 22.2.5
compute inverse of square triangular with nonzero diagonal entries	Examples 22.3.2, 22.3.3, 22.3.4
compute QR-decomposition via Gram–Schmidt on columns in invertible case	Examples 22.4.2, 22.4.3

Part V

Eigenvalues and second partial derivatives

"I developed my theory of infinitely many variables from purely mathematical interests, and even called it 'spectral analysis' without realizing it would later find an application to the actual spectrum of physics."

D. Hilbert [Hil]

"The Spectral Theorem is a glimpse at the source code of the universe."

anonymous former Math 51 student

Overview of Part V

In this final Part of the book, we introduce some new concepts that simultaneously extend and unify many of the topics we have been studying so far. This circle of ideas is centered on the notions of eigenvalue and of second derivatives of functions of several variables.

Chapters 23 and 24 introduce the fundamental concepts of eigenvalue and eigenvector of an $n \times n$ matrix, just scratching the surface of this vast subject. We mainly focus on *symmetric* $n \times n$ matrices, already a useful case for many applications and one for which things work out a bit more nicely than in general (largely due to the Spectral Theorem in Section 24.1 that is one of the most important results in applied linear algebra). Though we do not discuss general methods for computing eigenvectors in the main text (but we provide an introduction to those methods in Section E.5 for those who are interested), the exposure we provide in Chapters 23–24 will help you to get oriented with the basics.

For a function $f(x_1, \dots, x_n)$, Chapter 25 organizes its second-partials as the entries of an $n \times n$ matrix of functions, called the *Hessian matrix* $\mathrm{H}f$ of f. By the equality of mixed partials, this is actually a symmetric matrix; it provides the best language to describe the *quadratic* approximation to f near each critical point $\mathbf{a} \in \mathbf{R}^n$ (the quadratic polynomial in $x_1, \dots, x_n$ that best approximates f near $\mathbf{a}$).

We use the quadratic approximation to f at a critical point $\mathbf{a} \in \mathbf{R}^n$ in Chapter 26 to give a second-derivative test for determining when such an $\mathbf{a}$ is a local maximum or local minimum for f. This test relies on a certain property of the Hessian matrix $(\mathrm{H}f)(\mathbf{a})$ called its *definiteness*. A critical point $\mathbf{a}$ for f is a local minimum if $(\mathrm{H}f)(\mathbf{a})$ is positive-definite, and is a local maximum if $(\mathrm{H}f)(\mathbf{a})$ is negative-definite. Some critical points are saddle points, so the Hessian matrix there is not definite. We analyze the definiteness properties of $(\mathrm{H}f)(\mathbf{a})$ via the concepts of eigenvalue and eigenvector introduced in Chapter 23. Using that language, we characterize when the Hessian matrix is positive-definite or negative-definite, and when the critical point is a saddle point. This gives a rather detailed visualization of what a multivariable function looks like near a critical point.

In (the optional) Chapter 27 we first show how eigenvectors can be used to systematically "decouple" systems of ordinary differential equations, leading to explicit and exact solutions for these. Next, we describe an application to *population dynamics* (the study of how populations evolve when there is a fixed set of effects – births, deaths, mutations, etc. – from generation to generation), illustrating how questions in this direction often reduce to computing very high powers A^N of an $n \times n$ matrix A. Rather than multiplying the matrix A by itself a large number of times, which is a very computationally difficult problem, the powers A^N can be approximated rather effectively for large N in terms of the eigenvectors and eigenvalues of A. Finally, we describe the *singular value decomposition* (SVD) of a general $m \times n$ matrix, a more computationally stable replacement for the notion of eigenvalues for general (non-symmetric and non-square) matrices. SVD's are frequently encountered in practice, and we provide pointers to an array of applications.

23. Eigenvalues and eigenvectors

We now come to one of the most powerful concepts in linear algebra: eigenvalues and eigenvectors (special scalars and vectors associated to a square matrix). This will enable us to solve several problems encountered previously – for example, efficiently computing an $n \times n$ matrix raised to a big power. The short slogan is that *eigenvectors can be used to make an $n \times n$ matrix behave like a diagonal matrix.*

By the end of this chapter, you should be able to:

- check if a given $\mathbf{v} \in \mathbf{R}^n$ is an eigenvector of an $n \times n$ matrix A, and compute its eigenvalue if so;
- relate eigenvectors of A to null spaces of matrices related to A;
- for a 2×2 matrix, find eigenvalues and eigenvectors via an associated quadratic polynomial.

In practice, eigenvalues and eigenvectors for a non-triangular $n \times n$ matrix can be computed more reliably and quickly by a computer than by a human when $n > 2$. Therefore, you will not be asked to compute them from scratch in such cases when $n > 2$. Our emphasis is on how to use them for general n. (The optional Section E.5 introduces some ideas for finding eigenvalues and eigenvectors for general n.)

Eigenvalues grew out of the 19th-century efforts to clarify incomplete 18th-century work on the stability of solutions to differential equations in celestial mechanics [Ha1, pp. 563-565], and they arise in a broad range of contemporary disciplines. We saw in Chapter 16 that raising a matrix to a high power is relevant to predicting long-term behavior of systems that evolve in time (its relevance to why PageRank works is discussed in Remark D.2.1, resting on Theorem D.1.1), and we will see in Section 24.4 that eigenvalues control the behavior of this process. Other applications of eigenvalues include:

(i) (Example 24.6.2) the "principal axes" of a rotating rigid body (perhaps very irregularly-shaped!),
(ii) (Section 27.3) defining and computing the *singular value decomposition* of any matrix, a crucial tool across disciplines such as statistics, machine learning, financial forecasting, computer vision, computational biology, and data compression,
(iii) (Section 27.1) solving linked systems of "linear" differential equations, as arise for spring networks, coupled oscillators, and elsewhere in physical science (reaction rate problems in biochemistry involve "non-linear" systems, with long-term behavior approximated by linear systems).

Eigenvalues also show up in a vast array of other contexts, far too numerous to list here, such as:

- vibrational mechanics (e.g., dissipating energy in a way that gives stability); see Example 24.6.4,
- market risk analysis (extracting "principal components" from a correlation matrix of financial data over time to manage risk in portfolios; e.g., see [Al, Table I.2.3]),
- analysis of stability for (linear) electrical circuits (e.g., see [CDK, Sec. 10.5], where eigenvalues are called "natural frequencies").
- computing energies in quantum mechanics and physical chemistry (e.g., energy levels in atomic spectra are eigenvalues for a corresponding "Schrödinger matrix"; see Example 24.6.1).

In subsequent math courses (e.g., Math 53, 104, 113) and nearly any advanced course in the physical or computational sciences you will learn more about the tremendous usefulness of eigenvalues.

23.1. The main concepts.

Example 23.1.1. Consider $A = \begin{bmatrix} 16 & -2 & -6 \\ -2 & 19 & -3 \\ -6 & -3 & 27 \end{bmatrix}$. For $\mathbf{v} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$ we have (check!)

$$A\mathbf{v} = \begin{bmatrix} 24 \\ 12 \\ 12 \end{bmatrix} = 12\mathbf{v}.$$

So applying the linear transformation $T_A : \mathbf{R}^3 \to \mathbf{R}^3$ to $\mathbf{v}$ yields a multiple of $\mathbf{v}$ (with multiplier 12). ■

This situation in Example 23.1.1 is given a special name:

Definition 23.1.2. Let A be an $n \times n$ matrix. A vector $\mathbf{v} \in \mathbf{R}^n$ is called an *eigenvector* of A if it is **nonzero** and there is a scalar $\lambda \in \mathbf{R}$ for which $A\mathbf{v} = \lambda\mathbf{v}$. (Examples 23.1.5 and 23.1.6 give a nice visualization.) The prefix "eigen" is German[20] for "characteristic" or "own" (as in ownership).

The task of figuring out if a given $n \times n$ matrix A has *any* eigenvectors at all, and how to find them when it does, arises in many applications and is discussed in further math courses. We will always provide the eigenvectors to you when $n > 2$, as in Example 23.1.1 above. In Theorem 23.3.1 we will discuss the full story when $n = 2$ (and see Proposition 23.1.11 and Remark 23.1.13 for some more when $n > 2$).

Definition 23.1.3. If $\mathbf{v}$ is an eigenvector of an $n \times n$ matrix A, the scalar λ for which $A\mathbf{v} = \lambda\mathbf{v}$ is called the *eigenvalue*[21] of A associated with $\mathbf{v}$. The span of such a $\mathbf{v}$ is called an *eigenline* of A.

In materials science and other engineering contexts, the concepts of "natural frequencies" and "normal modes" correspond respectively to eigenvalues and eigenvectors in mathematical models.

Example 23.1.4. To illustrate these concepts in the case $n = 2$, consider the matrix $A = \begin{bmatrix} 3 & 16 \\ 2 & -1 \end{bmatrix}$. This has 7 and -5 as eigenvalues: for $\mathbf{v} = \begin{bmatrix} 4 \\ 1 \end{bmatrix}$ and $\mathbf{v}' = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$ you can (and should) check that

$$A\mathbf{v} = \begin{bmatrix} 28 \\ 7 \end{bmatrix} = 7\mathbf{v}, \quad A\mathbf{v}' = \begin{bmatrix} -10 \\ 5 \end{bmatrix} = (-5)\mathbf{v}'.$$

■

Let's prepare for a valuable visualization for the concepts of eigenvector and eigenvalue. If A has eigenvector $\mathbf{v}$, with associated eigenvalue λ, then any nonzero scalar multiple of $\mathbf{v}$ is also an eigenvector with the *same* eigenvalue. Indeed, if $\mathbf{w} = c\mathbf{v}$ (with $c \neq 0$, so $\mathbf{w} \neq \mathbf{0}$) then

$$A\mathbf{w} = A(c\mathbf{v}) = c(A\mathbf{v}) = c(\lambda\mathbf{v}) = \lambda(c\mathbf{v}) = \lambda\mathbf{w}. \tag{23.1.1}$$

Thus, *everything* in the line $\mathrm{span}(\mathbf{v})$ is multiplied by the scalar λ under the linear transformation for A.

Example 23.1.5. For an illustration with $n = 2$, consider the 2×2 matrix $A = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$. The effect of A on the points in a 2×2 grid centered at the origin is shown in Figure 23.1.1, in which the fat dots on the left are integer multiples of $\mathbf{u} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$. The vector $\mathbf{u}$ is an eigenvector of A with eigenvalue 1 and the vector $\mathbf{v}$ is an eigenvector of A with eigenvalue 3 since

$$A\mathbf{u} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \mathbf{u}, \quad A\mathbf{v} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 3 \\ -3 \end{bmatrix} = 3\mathbf{v}. \tag{23.1.2}$$

---

[20] The German words "eigenfunktion" and "eigenwert" were invented by David Hilbert in work on the mathematical methods of physics in the early 20th century. These were half-translated into English as "eigenfunction" and "eigenvalue" respectively, the former a precursor to "eigenvector" via an analogy between functions and vectors. The use of "eigen" won out in English over a zoo of competing terminologies (characteristic, secular, latent, proper) as explained here.

[21] The similarity with Lagrange multiplier notation is a coincidence. The concepts are unrelated, but there is a connection in a special case: Lagrange multipliers can be used in the proof of a deep result about eigenvalues in Chapter 24 (the Spectral Theorem) that underlies all of modern AI and machine learning (see "Method 1" in Step 3 near the end of Section B.3).

In Figure 23.1.1 this is illustrated by the red segment through $\mathbf{u}$ and $-\mathbf{u}$ being unaffected (multiplication by the eigenvalue 1 on $\mathrm{span}(\mathbf{u})$) and the blue segment through $\mathbf{v}$ and $-\mathbf{v}$ being stretched by a factor of 3 (multiplication by the eigenvalue 3 on $\mathrm{span}(\mathbf{v})$).

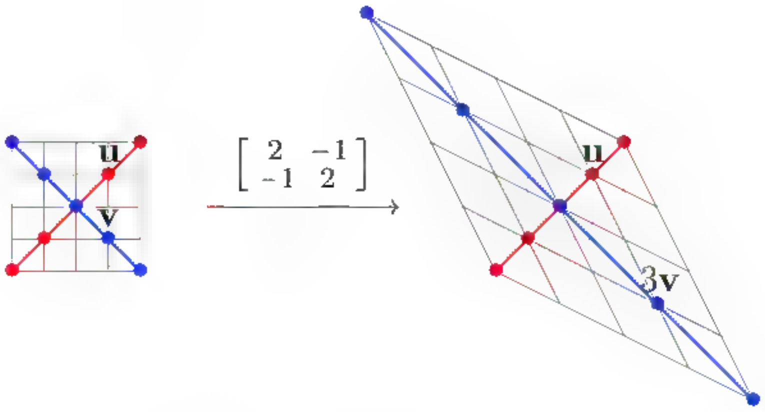


FIGURE 23.1.1. The vectors $\mathbf{u}$ and $\mathbf{v}$ are eigenvectors of the matrix A.

On the right side in Figure 23.1.1 we have superimposed the original 2×2 grid on top of its image under A to illustrate that the red segment really has not changed whereas there has been a lot of stretching along the $\pm\mathbf{v}$-directions (the fat dots are integer multiples of $\mathbf{u}$ and $3\mathbf{v}$). ■

Example 23.1.6. The matrix $M = \begin{bmatrix} 5/3 & -2/3 \\ -1/3 & 4/3 \end{bmatrix}$ has $\mathbf{u} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ as an eigenvector with eigenvalue 1 and $\mathbf{v} = \begin{bmatrix} 1 \\ -1/2 \end{bmatrix}$ as an eigenvector with eigenvalue 2. This is illustrated in Figure 23.1.2 (in which the fat dots on the left are integer multiples of $\mathbf{u}$ and $\mathbf{v}$). In contrast with Example 23.1.5, now the eigenvectors $\mathbf{u}$ and $\mathbf{v}$ are *not orthogonal* (the angle between them is $\arccos(1/\sqrt{10}) \approx 71.6°$).

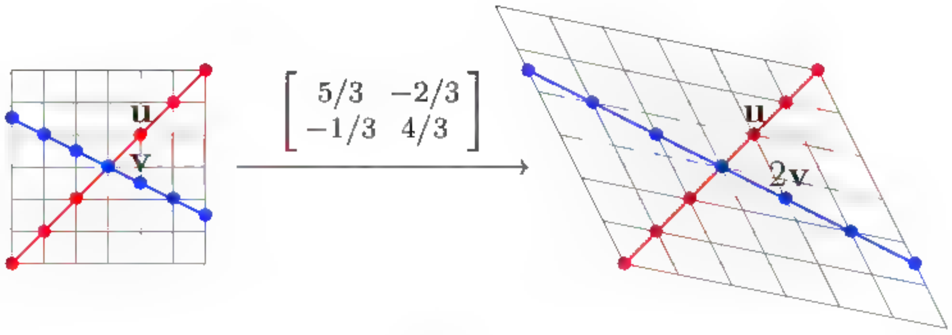


FIGURE 23.1.2. The vectors $\mathbf{u}$ and $\mathbf{v}$ are eigenvectors of the matrix M.

The failure of orthogonality for eigenvectors of M is related to M *not* being symmetric (in the sense of Definition 20.3.5) whereas A in Example 23.1.5 is symmetric; we'll return to this in Section 24.1. ■

We reiterate for emphasis: if $\mathbf{v}$ is an eigenvector of an $n \times n$ matrix A with eigenvalue λ, the effect of the linear transformation $T_A : \mathbf{R}^n \to \mathbf{R}^n$ on *all* points belonging to the line spanned by $\mathbf{v}$ is scaling by the factor λ. This is a good way to think about the geometric meaning of eigenvectors: they provide lines through $\mathbf{0}$ on which the effect of T_A is scaling by some multiplier factor $\lambda \in \mathbf{R}$ (so stretching if $|\lambda| > 1$, shrinking if $|\lambda| < 1$, and flipping across the origin if $\lambda < 0$). This visualization has a useful special case:

Example 23.1.7. For an angle θ that is neither 0° nor 180°, consider the rotation matrix

$$A_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

(which is not symmetric). Then for any nonzero $\mathbf{v} \in \mathbf{R}^2$, the vector $A_\theta \mathbf{v}$ is at an angle $\theta \neq 0^\circ, 180^\circ$ relative to $\mathbf{v}$ and hence is *not* on the same line as that spanned by $\mathbf{v}$. Thus, $A_\theta \mathbf{v}$ is not a scalar multiple of $\mathbf{v}$, so $\mathbf{v}$ cannot be an eigenvector for A_θ. Hence, A_θ *does not have any eigenvectors at all* in $\mathbf{R}^2$. ■

Remark 23.1.8 (online resource). Some instructive dynamic visualizations for the ideas of eigenvalue and eigenvector are given in this video at "Essence of Linear Algebra".

Example 23.1.9. If the null space of an $n \times n$ matrix A contains a nonzero vector $\mathbf{w}$ then $A\mathbf{w} = \mathbf{0} = 0\mathbf{w}$, so $\mathbf{w}$ is an eigenvector of A with eigenvalue 0. An eigenvector for A with eigenvalue 0 is the same thing as a nonzero vector in the null space $N(A)$, so 0 is an eigenvalue of A *precisely when* A is not invertible (by Theorem 18.3.3).

Warning. Even though the scalar 0 might occur as an eigenvalue, don't forget that *by definition* the zero vector $\mathbf{0}$ is *forbidden* from being considered as an eigenvector (even though it satisfies $A\mathbf{0} = \lambda\mathbf{0}$ for every λ). This is important to avoid a lot of confusion later on. Many results about eigenvectors break down if we allow $\mathbf{0}$ to be considered as one (forcing us to keep saying "nonzero" all the time, which would get cumbersome as we do ever more with eigenvectors); this is the main reason that the possibility is ruled out by requiring eigenvectors to be nonzero by *definition*. Further experience will convince you of the aptness of this convention if you are bothered by it now. ■

Remark 23.1.10. To illustrate the "Warning" above, let's explain why for an $n \times n$ matrix A and a given eigenvector $\mathbf{v}$ for A, there cannot be scalars $\lambda \neq \mu$ for which $A\mathbf{v} = \lambda\mathbf{v}$ and $A\mathbf{v} = \mu\mathbf{v}$ (i.e., Definition 23.1.3 is unambiguous: we can speak of "the" eigenvalue associated to a given eigenvector). The reason is that if $A\mathbf{v} = \lambda\mathbf{v}$ and $A\mathbf{v} = \mu\mathbf{v}$ for scalars $\lambda, \mu \in \mathbf{R}$ then $\lambda\mathbf{v} = A\mathbf{v} = \mu\mathbf{v}$, so $\lambda\mathbf{v} = \mu\mathbf{v}$ and hence by subtraction

$$(\lambda - \mu)\mathbf{v} = \mathbf{0}.$$

The vector $\mathbf{v}$ is **nonzero** because eigenvectors are nonzero *by definition*, and the only scalar that multiplies a nonzero vector to yield the zero vector is the scalar 0. Thus, $\lambda - \mu = 0$, so $\lambda = \mu$ as desired.

On the other hand, for a given eigenvalue λ of A there may be lots of linearly independent eigenvectors with associated eigenvalue λ! For instance, in Example 23.2.3 we give a 3×3 matrix A having linearly independent eigenvectors $\mathbf{v}, \mathbf{v}' \in \mathbf{R}^3$ each with eigenvalue 2 (i.e., $A\mathbf{v} = 2\mathbf{v}$ and $A\mathbf{v}' = 2\mathbf{v}'$). Or as an extreme case, the identity matrix I_n has *every* nonzero n-vector $\mathbf{v}$ as an eigenvector with eigenvalue $\lambda = 1$.

In practice, one *first* figures out the eigenvalues of an $n \times n$ matrix A, and *then* works out eigenvectors for each eigenvalue by computing a null space. To explain this, recall that a scalar λ is an eigenvalue of A when the vector equation $A\mathbf{x} = \lambda\mathbf{x}$ has a *nonzero* solution $\mathbf{x} \in \mathbf{R}^n$. But this is the same as $A\mathbf{x} - \lambda\mathbf{x} = \mathbf{0}$, and $A\mathbf{x} - \lambda\mathbf{x} = A\mathbf{x} - \lambda\mathrm{I}_n\mathbf{x} = (A - \lambda\mathrm{I}_n)\mathbf{x}$. Thus, an eigenvector with eigenvalue λ is a *nonzero* vector in the null space $N(A - \lambda\mathrm{I}_n)$. But an $n \times n$ matrix M (such as $A - \lambda\mathrm{I}_n$) has a nonzero vector in its null space precisely when M is not invertible (Theorem 18.3.3). Thus:

Proposition 23.1.11. A scalar λ is an eigenvalue for an $n \times n$ matrix A precisely when $A - \lambda\mathrm{I}_n$ is not invertible, or equivalently the null space $N(A - \lambda\mathrm{I}_n)$ contains a nonzero vector. The eigenvectors for A with eigenvalue λ are the nonzero vectors in $N(A - \lambda\mathrm{I}_n)$ (which is called the λ-*eigenspace* for A).

Example 23.1.12. Every $n \times n$ Markov matrix M has 1 as an eigenvalue; i.e., there is a nonzero n-vector $\mathbf{v}$ for which $M\mathbf{v} = \mathbf{v}$. This has a probabilistic meaning explored in Chapter 16 when M^k approaches a matrix whose columns all approach a common n-vector as $k \to \infty$, which we saw happens in some cases (such as bird migrations and PageRank) but not in others (such as gambler's ruin).

To see that 1 is an eigenvalue of any such M, by Proposition 23.1.11 we want to show the null space $N(M - \mathrm{I}_n)$ is nonzero, or in other words has dimension at least 1. By the Rank–Nullity Theorem (Theorem

21.3.16) we have $\dim C(M - \mathrm{I}_n) + \dim N(M - \mathrm{I}_n) = n$, so the desired property $\dim N(M - \mathrm{I}_n) \geq 1$ is the same as the property $\dim C(M - \mathrm{I}_n) \leq n - 1$. This says that the subspace $C(M - \mathrm{I}_n) \subset \mathbf{R}^n$ is not the entirety of $\mathbf{R}^n$. We'll show all vectors in $C(M - \mathrm{I}_n)$ have entries summing to 0, which will do the job.

By definition of M being Markov, its columns sum to 1. The columns of I_n each sum to 1 also. Hence, the columns of $A = M - \mathrm{I}_n$ each sum to $1 - 1 = 0$. Every vector in $C(A)$ is a linear combination $A\mathbf{x} = \sum_{j=1}^{n} x_j A\mathbf{e}_j$ of the columns $A\mathbf{e}_j$ of A, so

$$\text{sum of entries in } A\mathbf{x} = \sum_{j=1}^{n} x_j(\text{sum of entries in } A\mathbf{e}_j) = \sum_{j=1}^{n} x_j 0 = 0. \qquad \blacksquare$$

Remark 23.1.13. In Section 23.3 we will use Proposition 23.1.11 for $n = 2$ via the characterization of non-invertibility of 2×2 matrices in terms of vanishing of the determinant: $\det(A - \lambda\mathrm{I}_2) = 0$. By Remark 18.2.5, non-invertibility of an $n \times n$ matrix M for general n is characterized by the vanishing of $\det(M)$, so eigenvalues λ of an $n \times n$ matrix A are solutions to $\det(A - \lambda\mathrm{I}_n) = 0$. This determinant turns out to be a polynomial in λ with degree n, determined by the entries in A (discussed in Section E.5 for those who are interested; the case $n = 2$ is treated in Section 23.3). So for n not too big, to find the eigenvalues of A in $\mathbf{R}$ just ask a computer where the graph of that degree-n polynomial crosses the x-axis.

23.2. Diagonal and triangular matrices.

Example 23.2.1. There is one general class of matrices for which it is easy to read off some eigenvectors and the corresponding eigenvalues: diagonal matrices. If A is a diagonal $n \times n$ matrix $\begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix}$ then the ith standard basis vector $\mathbf{e}_i$ is an eigenvector of A with eigenvalue d_i: $A\mathbf{e}_i = d_i\mathbf{e}_i$

This is easiest to see by working out a typical 3×3 example and observing the pattern that emerges (which has nothing to do with the specific diagonal entries, nor the fact that $n = 3$): for

$$A = \begin{bmatrix} 7 & 0 & 0 \\ 0 & -4 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

we can verify by direct calculation (check) that $A\mathbf{e}_1 = 7\mathbf{e}_1$, $A\mathbf{e}_2 = -4\mathbf{e}_2$, $A\mathbf{e}_3 = 5\mathbf{e}_3$. For example:

$$A\mathbf{e}_2 = \begin{bmatrix} 7 & 0 & 0 \\ 0 & -4 & 0 \\ 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -4 \\ 0 \end{bmatrix} = -4\mathbf{e}_2,$$

and the cases of $\mathbf{e}_1$ and $\mathbf{e}_3$ work out similarly (with respective multipliers 7 and 5 from the corresponding diagonal entries of A).

On the other hand, *there are no other eigenvalues* for this A. Indeed, for $\lambda \neq 7, -4, 5$ the equation $A\mathbf{x} = \lambda\mathbf{x}$ expresses the simultaneous conditions

$$7x_1 = \lambda x_1, \quad -4x_2 = \lambda x_2, \quad 5x_3 = \lambda x_3$$

with $\lambda \neq 7, -4, 5$. This forces all x_i's to vanish (so $\mathbf{x} = \mathbf{0}$ and hence there is no nonzero solution, so λ isn't an eigenvalue of A); e.g., if $x_2 \neq 0$ then we could cancel it in the second equation to get $\lambda = -4$, which is false, so necessarily $x_2 = 0$, and similarly x_1 and x_3 must vanish since $\lambda \neq 7, 5$ respectively.

This argument applies equally well to any diagonal matrix: its *only* eigenvalues are the diagonal entries. $\blacksquare$

The theory of eigenvectors gives a way to make a lot of $n \times n$ matrices A "look diagonal" when the effect of the corresponding linear transformation is expressed in an appropriate reference frame (typically quite different from the standard basis!). More precisely, *if* there is a basis $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ of $\mathbf{R}^n$ consisting of eigenvectors for A, say with respective eigenvalues $\lambda_1, \dots, \lambda_n$ (some of which may be equal to each other), then the effect of A "looks diagonal" when n-vectors $\mathbf{x}$ are expressed in terms of this basis.

For example, if we consider $M = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$ then in Example 23.1.5 we saw that the basis of $\mathbf{R}^2$ given by $\mathbf{u} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ consists of eigenvectors for M with respective eigenvalues 1 and 3, so for $\mathbf{x} = c_1\mathbf{u} + c_2\mathbf{v}$ we have $M\mathbf{x} = M(c_1\mathbf{u} + c_2\mathbf{v}) = c_1(M\mathbf{u}) + c_2(M\mathbf{v}) = c_1\mathbf{u} + c_2(3\mathbf{v}) = c_1\mathbf{u} + (3c_2)\mathbf{v}$.

More generally, if we write $\mathbf{x} = c_1\mathbf{v}_1 + \cdots + c_n\mathbf{v}_n$ for a basis of eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_n$ (when such a basis exists) with $A\mathbf{v}_j = \lambda_j\mathbf{v}_j$ then

$$A\mathbf{x} = c_1 A\mathbf{v}_1 + \cdots + c_n A\mathbf{v}_n = c_1\lambda_1\mathbf{v}_1 + \cdots + c_n\lambda_n\mathbf{v}_n = \lambda_1(c_1\mathbf{v}_1) + \cdots + \lambda_n(c_n\mathbf{v}_n).$$

So in words: if $\mathbf{x}$ has coefficients $(c_1, \dots, c_n)$ in terms of the $\mathbf{v}_j$'s then $A\mathbf{x}$ has coefficients $(\lambda_1 c_1, \dots, \lambda_n c_n)$ in terms of the $\mathbf{v}_j$'s. Thus, the effect of the matrix-vector product against A behaves like the diagonal matrix of λ_j's when general n-vectors are *written in terms of the basis of eigenvectors* $\mathbf{v}_1, \dots, \mathbf{v}_n$ for A!

You might think that this is of limited utility because it must be a very restrictive and difficult-to-check condition on A that $\mathbf{R}^n$ has a basis consisting of eigenvectors for A. One of the truly great theorems of mathematics is that for symmetric matrices such a basis always exists! (Such bases also exist for some non-symmetric matrices, and fail to exist for others; see Examples 23.2.3 and 23.3.6.) We will discuss this amazing result and its awesome applications in the remaining chapters of this book.

The preceding analysis of eigenvalues and eigenvectors for diagonal $n \times n$ matrices can be pushed further, to the cases of upper triangular and lower triangular $n \times n$ matrices:

Theorem 23.2.2. Let M be an $n \times n$ upper triangular or lower triangular matrix. The eigenvalues of M are exactly the diagonal entries. For each eigenvalue λ the corresponding eigenvectors are the nonzero vectors in the null space $N(M - \lambda \mathrm{I}_n)$: nonzero solutions $\mathbf{x}$ to the (upper or lower) triangular system $(M - \lambda \mathrm{I}_n)\mathbf{x} = \mathbf{0}$ of n linear equations in n unknowns (which we can solve via back-substitution!).

Why are the eigenvalues of an $n \times n$ triangular matrix M precisely the diagonal entries? Letting $d_i = m_{ii}$ denote the ith diagonal entry, $M - \lambda \mathrm{I}_n$ is also a triangular matrix with diagonal entries $d_i - \lambda$. It is a general fact (verified by direct computation when $n = 2$) that the determinant of a triangular matrix is equal to the product of its diagonal entries, so $\det(M - \lambda \mathrm{I}_n) = (d_1 - \lambda)(d_2 - \lambda)\cdots(d_n - \lambda)$. This vanishes precisely when λ is equal to one of the d_i's, as desired.

Let's illustrate finding eigenspaces via back-substitution for a 3×3 upper triangular matrix:

Example 23.2.3. Let $M = \begin{bmatrix} 2 & 3 & 1 \\ 0 & 8 & 2 \\ 0 & 0 & 2 \end{bmatrix}$. This has as its eigenvalues 2 and 8. We work out the corresponding eigenvectors by solving the corresponding vector equations $M\mathbf{x} = 2\mathbf{x}$ and then $M\mathbf{x} = 8\mathbf{x}$, or equivalently $(M - 2\mathrm{I}_3)\mathbf{x} = \mathbf{0}$ and then $(M - 8\mathrm{I}_3)\mathbf{x} = \mathbf{0}$. These are each upper triangular systems of linear equations, which we know how to solve via back substitution.

First consider $(M - 2\mathrm{I}_3)\mathbf{x} = \mathbf{0}$. This is the system of equations

$$\begin{aligned} 3x_2 + x_3 &= 0 \\ 6x_2 + 2x_3 &= 0 \\ 0 &= 0 \end{aligned}$$

for which the last equation says nothing and the first two each say $x_3 = -3x_2$. There is no condition on x_1, so the solutions are all vectors of the form

$$\begin{bmatrix} x_1 \\ x_2 \\ -3x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ x_2 \\ -3x_2 \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \\ -3 \end{bmatrix}.$$

Hence, $N(M - 2\mathrm{I}_3)$ is 2-dimensional, with basis $\mathbf{v} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{v}' = \begin{bmatrix} 0 \\ 1 \\ -3 \end{bmatrix}$. Every linear combination $a\mathbf{v} + b\mathbf{v}'$ with a, b not both 0 is then an eigenvector for M with eigenvalue 2. (If the $(2,3)$-entry of M were anything other than 2, the outcome of this calculation would have been that $x_2, x_3 = 0$, which is to say that we would have obtained that $N(M - 2\mathrm{I}_3)$ is the line spanned by $\mathbf{e}_1$.)

Next, we consider $(M - 8\mathrm{I}_3)\mathbf{x} = \mathbf{0}$. This is the system of equations

$$\begin{aligned} -6x_1 + 3x_2 + x_3 &= 0 \\ 2x_3 &= 0 \\ -6x_3 &= 0 \end{aligned}$$

for which the final two equations say $x_3 = 0$ and so the first equation says $-6x_1 + 3x_2 = 0$. In other words, $x_3 = 0$ and $x_2 = 2x_1$, so the solutions are the vectors of the form

$$\begin{bmatrix} x_1 \\ 2x_1 \\ 0 \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}.$$

The eigenvectors of M with eigenvalue 8 are precisely the nonzero scalar multiples of $\begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}$. ■

23.3. The case $n = 2$. It is a pain to compute eigenvectors and eigenvalues for an $n \times n$ matrix in general, but for $n = 2$ the computation can be related to familiar tasks from algebra: the quadratic formula and equations of lines. We illustrate this by revisiting the 2×2 matrix

$$A = \begin{bmatrix} 3 & 16 \\ 2 & -1 \end{bmatrix}$$

from Example 23.1.4 to explain how to find the eigenvectors and eigenvalues in this case, and then we formulate the conclusion in a manner that applies to any 2×2 matrix whatsoever. We'll see that the essential issue is really *first to figure out the possibilities for the eigenvalues*, which will amount to using the quadratic formula (and then finding corresponding eigenvectors will turn out to be quite easy). For $n > 2$ there is nothing quite like the quadratic formula.

Remark 23.1.13 tells us how to find the eigenvalues: we seek solutions to $\det(A - \lambda\mathrm{I}_2) = 0$. To compute this determinant, we write out the 2×2 matrix in full glory:

$$A - \lambda\mathrm{I}_2 = \begin{bmatrix} 3 & 16 \\ 2 & -1 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 3 - \lambda & 16 \\ 2 & -1 - \lambda \end{bmatrix},$$

so its determinant is

$$(3-\lambda)(-1-\lambda) - 16(2) = -3 - 2\lambda + \lambda^2 - 32 = \lambda^2 - 2\lambda - 35.$$

This is a quadratic polynomial in λ!

We can solve this quadratic condition using the quadratic formula, getting that it has as its solutions

$$\frac{2 \pm \sqrt{4 + 4 \cdot 35}}{2} = \frac{2 \pm \sqrt{144}}{2} = \frac{2 \pm 12}{2} = 1 \pm 6 = 7, -5$$

(or in this particular case we might happen to notice that the quadratic polynomial factors as $(\lambda-7)(\lambda+5)$, again yielding its roots as 7 and -5). For these two eigenvalues, the eigenvectors are *nonzero* solutions $\mathbf{x} \in \mathbf{R}^2$ to the vector equation $A\mathbf{x} = 7\mathbf{x}$ when $\lambda = 7$ and $A\mathbf{x} = -5\mathbf{x}$ when $\lambda = -5$.

Writing $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$, the condition $A\mathbf{x} = 7\mathbf{x}$ amounts to the system of equations

$$3x + 16y = 7x, \;\; 2x - y = 7y,$$

and by bringing the right side over to the left side this becomes

$$-4x + 16y = 0, \;\; 2x - 8y = 0.$$

These equations are multiples of each other, and any nonzero vector on this common line is an eigenvector with eigenvalue 7. Writing the line as $y = (1/4)x$, a clean option is $(x, y) = (4, 1)$, recovering $\mathbf{v} = \begin{bmatrix} 4 \\ 1 \end{bmatrix}$ that we encountered in Example 23.1.4. (We could have chosen any other nonzero value $y = c$, so $x = 4c$, yielding $\mathbf{x} = \begin{bmatrix} 4c \\ c \end{bmatrix} = c\mathbf{v}$ as an eigenvector for the eigenvalue 7.) Note that it is no surprise that we obtained two scalar equations that are scalar multiples of each other, since 7 being an eigenvalue ensures that $A\mathbf{x} = 7\mathbf{x}$ has at least a line through the origin among its solutions (namely, the scalar multiples of an eigenvector with eigenvalue 7).

The story goes similarly for $\lambda = -5$: the vector equation $A\mathbf{x} = -5\mathbf{x}$ amounts to the system of equations

$$3x + 16y = -5x, \;\; 2x - y = -5y,$$

which can be rewritten as

$$8x + 16y = 0, \;\; 2x + 4y = 0.$$

These are multiples of each other, and any nonzero vector on this common line is an eigenvector with eigenvalue -5. Writing the line as $y = -(1/2)x$, a clean option is $(x, y) = (-2, 1)$, or perhaps $(2, -1)$. The latter choice recovers the eigenvector $\mathbf{v}' = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$ with eigenvalue -5 seen in Example 23.1.4.

We now present the conclusion in a general form that can be readily applied to any 2×2 matrix. The verification of the general formulation below is essentially just a generalization of the specific calculations given above upon making the general calculation

$$\begin{aligned} \det(A - \lambda \mathrm{I}_2) = \det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}\right) = \det\left(\begin{bmatrix} a-\lambda & b \\ c & d-\lambda \end{bmatrix}\right) &= (a-\lambda)(d-\lambda) - bc \\ &= \lambda^2 - (a+d)\lambda + (ad - bc); \end{aligned}$$

see Section 23.4 for such details. (If you take Math 53, Math 104, or Math 113 you will learn about the additional ideas that adapt the preceding considerations to $n > 2$; see Theorem E.5.1 and Remark E.5.2 in Appendix E for an introduction to the case of general n.)

Theorem 23.3.1. Let $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ be a 2×2 matrix. Define its *trace* $\mathrm{tr}(A) = a + d$ to be the sum of its diagonal entries, and its *determinant* $\det(A) = ad - bc$ to be the difference of the products of entries on the diagonal and the anti-diagonal.

(i) The eigenvalues of A in $\mathbf{R}$ are exactly the roots of $P_A(\lambda) = \lambda^2 - \mathrm{tr}(A)\lambda + \det(A)$. (This quadratic polynomial is called the *characteristic polynomial* of A.)
(ii) For each such root λ, the corresponding eigenvectors are the nonzero vectors in $N(A - \lambda \mathrm{I}_2)$.

Corollary 23.3.2. A 2×2 matrix A has an eigenvalue precisely when $P_A(\lambda)$ has a real root, which is to say that its discriminant $\mathrm{tr}(A)^2 - 4\det(A)$ is ≥ 0.

When P_A has a pair of roots $\lambda_1, \lambda_2 \in \mathbf{R}$ then the relationship between the roots and the coefficients of a quadratic polynomial $(X - r)(X - s) = X^2 - (r + s)X + rs$ gives that the product $\lambda_1\lambda_2$ of those eigenvalues is equal to the constant term $\det(A)$ of P_A and the sum $\lambda_1 + \lambda_2$ of those eigenvalues is equal to $\mathrm{tr}(A)$ (negative of the linear coefficient).

Remark 23.3.3. For each eigenvalue λ, when expressing "$A\mathbf{x} = \lambda\mathbf{x}$" as a pair of scalar equations we know that this must have a nonzero solution and hence *cannot* correspond to the equations of two distinct lines through the origin (whose only solution is $\mathbf{0}$). If you get something other than scalar multiples of the same equation twice then you have made a mistake! (In particular, if you pick any λ that isn't an eigenvalue and try to compute $N(A - \lambda\mathrm{I}_2)$, you'll always get $\{\mathbf{0}\}$.)

Here are a few examples to illustrate Theorem 23.3.1 and Corollary 23.3.2.

Example 23.3.4. Let's once again revisit the matrix

$$A = \begin{bmatrix} 3 & 16 \\ 2 & -1 \end{bmatrix}$$

considered in the preceding discussion.

We directly compute $\mathrm{tr}(A) = 3 - 1 = 2$ and $\det(A) = 3(-1) - 16 \cdot 2 = -3 - 32 = -35$, so the eigenvalues of A are precisely the roots of the quadratic polynomial $\lambda^2 - 2\lambda - 35$. That is exactly the quadratic polynomial that we arrived at above, and its roots can be found using the quadratic formula (or observation of a factorization as $(\lambda - 7)(\lambda + 5)$): 7 and -5. Each of $A\mathbf{x} = 7\mathbf{x}$ and $A\mathbf{x} = -5\mathbf{x}$ written as a pair of linear equations in 2 unknowns is found to be scalar multiples of a single equation, which is to say the same line written twice, and we can write down a nonzero vector on each such line as we did above. ■

Example 23.3.5. Consider the matrix $A = \begin{bmatrix} 1 & 2 \\ 2 & -2 \end{bmatrix}$. In this case $\mathrm{tr}(A) = 1 - 2 = -1$ and $\det(A) = 1(-2) - 2 \cdot 2 = -2 - 4 = -6$, so $P_A(\lambda) = \lambda^2 - (-1)\lambda - 6 = \lambda^2 + \lambda - 6$. Using the quadratic formula or observing the factorization $(\lambda + 3)(\lambda - 2)$ yields two roots: -3 and 2.

To find corresponding eigenvectors, we write each of the conditions $A\mathbf{x} = -3\mathbf{x}$ and $A\mathbf{x} = 2\mathbf{x}$ as a pair of simultaneous linear equations in 2 unknowns, and in each case the pair of equations will be multiples of each other and so correspond to a single line. Anything nonzero on that line is an eigenvector for the corresponding eigenvalue.

Beginning with the eigenvalue 2, for $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$ the vector equation $A\mathbf{x} = 2\mathbf{x}$ is the pair of simultaneous equations

$$x + 2y = 2x, \quad 2x - 2y = 2y,$$

which we can rewrite by bringing the right side over to the left side in each equation:

$$-x + 2y = 0, \;\; 2x - 4y = 0.$$

These equations are multiples of each other (as we know they have to be), corresponding to the common line $y = (1/2)x$. One nonzero point on this line is $(2, 1)$, yielding the eigenvector $\mathbf{w}_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$. (We could have chosen many other nonzero points on this line, getting many other eigenvectors, all nonzero scalar multiples of $\mathbf{w}_1$: the point $(6, 3)$ yields $\begin{bmatrix} 6 \\ 3 \end{bmatrix} = 3\mathbf{w}_1$, the point $(-5, -5/2)$ yields $\begin{bmatrix} -5 \\ -5/2 \end{bmatrix} = -(5/2)\mathbf{w}_1$, etc.)

Next turning to the eigenvalue -3, for $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$ the vector equation $A\mathbf{x} = -3\mathbf{x}$ is the pair of simultaneous equations

$$x + 2y = -3x, \;\; 2x - 2y = -3y,$$

which we can rewrite as

$$4x + 2y = 0, \;\; 2x + y = 0.$$

Once again, these equations are multiples of each other (as they must be), corresponding to the common line $y = -2x$. A nonzero point on this line is $(-1, 2)$, yielding the eigenvector $\mathbf{w}_2 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$. ■

Example 23.3.6. In Example 23.1.7 we saw via geometric reasoning that the 2×2 rotation matrix A through an angle $\theta \neq 0^\circ, 180^\circ$ has no eigenvalues at all. In terms of Corollary 23.3.1, the quadratic polynomial $P_A(\lambda)$ turns out to have discriminant < 0 in such cases, so there are no real roots. Let's see this in action for a specific example, with the angle $\theta = 30^\circ$. In this case

$$A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} \sqrt{3}/2 & -1/2 \\ 1/2 & \sqrt{3}/2 \end{bmatrix},$$

so $\mathrm{tr}(A) = \sqrt{3}$ and $\det(A) = 3/4 - (-1/4) = 1$. Hence, $P_A(\lambda) = \lambda^2 - \sqrt{3}\lambda + 1$, whose discriminant is $(\sqrt{3})^2 - 4 \cdot 1 = 3 - 4 = -1 < 0$, so indeed there are no real roots (and so no eigenvalues in $\mathbf{R}$). ■

Remark 23.3.7. For some applications of $n \times n$ matrices A with truly gigantic n (as arise when using singular value decomposition and principal component analysis, techniques introduced in Section 27.3 that are fundamental in data analysis), one needs to compute the several largest eigenvalues of A. Fortunately, there are algorithms to achieve this (especially for extremely large n) without having to compute all of the eigenvalues.

Sometimes with n not so gigantic, one needs to analyze all of the eigenvalues (see Section J.1 for such a situation in chemistry). For $n = 2$, we have seen how to do this using a certain quadratic polynomial, in Theorem 23.3.1. That technique adapts to general n, as is discussed in Section E.5, and is useful in the theoretical development of linear algebra (e.g., to prove additional general results about eigenvalues) and in some applications (such as in the analysis of algorithms).

However, for accurate numerical computation of eigenvalues of a given $n \times n$ matrix, the approach via roots of a specific polynomial is not so useful once n is bigger than 3. Appendix H discusses an amazing procedure called the *QR algorithm* (unsurprisingly from the name, it is based on the QR-decomposition) that is the means by which the full set of eigenvalues of square matrices is actually computed to high accuracy on a computer. It has been called "one of the jewels in the crown of matrix computations" [**Hig**], and in January 2000 it appeared in the (chronologically ordered) list "Top Ten Algorithms of the (20th) Century" [**DS**]. (Two other entries on the list were the "Simplex Method for Linear Programming", which we briefly discussed near the end of Section 10.4, and "The Decomposition Approach to Matrix Computations" for which Chapter 22 gave an introduction.)

23.4. Proofs about the characteristic polynomial. This section discusses proofs of Theorem 23.3.1 and Corollary 23.3.2. First, we prove Theorem 23.3.1, using notation as in its statement.

PROOF. To prove (i), we have to show for a given scalar λ that the vector equation $A\mathbf{x} = \lambda\mathbf{x}$ has a *nonzero* solution $\mathbf{x} \in \mathbf{R}^2$ precisely when $P_A(\lambda) = 0$. The eigenvector condition $A\mathbf{x} = \lambda\mathbf{x}$ says exactly that $(A - \lambda \mathrm{I}_2)\mathbf{x} = \mathbf{0}$. Since

$$A - \lambda\mathrm{I}_2 = \begin{bmatrix} a & b \\ c & d \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} a-\lambda & b \\ c & d-\lambda \end{bmatrix},$$

λ is an eigenvalue exactly when this final matrix has a nonzero null space. Observe that this final matrix has determinant equal to

$$(a-\lambda)(d-\lambda) - bc = ad - (a+d)\lambda + \lambda^2 - bc = \lambda^2 - (a+d)\lambda + (ad-bc) = P_A(\lambda),$$

so our task is to show $A - \lambda\mathrm{I}_2$ has a nonzero null space precisely when its determinant vanishes.

In other words, (i) comes down to proving a general fact about 2×2 matrices having nothing to do with eigenvalues: a general 2×2 matrix M has nonzero null space precisely when $\det(M) = 0$. Let's first assume that there is a nonzero null space and aim to deduce that $\det(M) = 0$. If either column vanishes then certainly $\det(M) = 0$ (straight from the definition of $\det(M)$), so we can suppose both columns are not the zero vector in $\mathbf{R}^2$.

A nonzero vector $\mathbf{x} = \left[\begin{smallmatrix} u \\ v \end{smallmatrix}\right]$ in the null space of M expresses a linear dependence relation on the columns of M: the vector $M\mathbf{x}$ is u times the first column $\mathbf{m}_1$ plus v times the second column $\mathbf{m}_2$ (see Theorem 13.4.1). At least one of u or v is nonzero, so by Theorem 19.1 this says that the columns $\mathbf{m}_1$ and $\mathbf{m}_2$ are linearly dependent, or in other words that one of the $\mathbf{m}_i$'s is a scalar multiple of the other. We arranged that both of these columns are nonzero, so the scalar multiplier must be nonzero and hence can be divided to the other side if necessary to write the relation among the columns as $\mathbf{m}_2 = t\mathbf{m}_1$ for some (nonzero) scalar t. This vector equation says $b = ta$ and $d = tc$, so

$$\det(M) = ad - bc = a(tc) - (ta)c = 0.$$

That is, we have shown that if M has a nonzero null space then $\det(M) = 0$.

In the reverse direction, if $\det(M) = 0$ (that is, if $ad = bc$) then we want to show that M has a nonzero null space. The given relation $ad = bc$ yields by direct calculation that each of the following vectors is in the null space of M:

$$\begin{bmatrix} d \\ -c \end{bmatrix}, \begin{bmatrix} b \\ -a \end{bmatrix}.$$

As long as one of these is not the zero vector, we win. But if both are the zero vector then M is the zero matrix, so *everything* in $\mathbf{R}^2$ is in the null space of M (so the null space is still nonzero). This completes the proof of (i).

For an eigenvalue λ, by definition the null space of $A - \lambda\mathrm{I}_2$ is nonzero and the eigenvectors with eigenvalue λ are precisely the nonzero vectors in this null space (since $(A - \lambda\mathrm{I}_2)\mathbf{x} = A\mathbf{x} - \lambda\mathrm{I}_2\mathbf{x} = A\mathbf{x} - \lambda\mathbf{x}$ vanishes precisely when $A\mathbf{x} = \lambda\mathbf{x}$). This establishes (ii). ☐

The assertions in Corollary 23.3.2 express general facts about quadratic polynomials $X^2 - tX + d$: by the quadratic formula there is a root in $\mathbf{R}$ precisely when the discriminant $(-t)^2 - 4d = t^2 - 4d$ is non-negative, and in such cases with roots λ_1, λ_2 we have the factorization

$$X^2 - tX + d = (X - \lambda_1)(X - \lambda_2) = X^2 - (\lambda_1 + \lambda_2)X + \lambda_1\lambda_2.$$

Comparing coefficients, this says $t = \lambda_1 + \lambda_2$ and $d = \lambda_1\lambda_2$. Hence, Corollary 23.3.2 is an immediate consequence of Theorem 23.3.1.

Chapter 23 highlights (links to highlights in and next chapters)

Notation	**Meaning**	**Location in text**
μ	Greek analogue of lower-case "em"; read as "mew"	Remark 23.1.10
$\mathrm{tr}(A)$ (for 2×2 matrix A)	trace of A	Theorem 23.3.1
$P_A(\lambda)$ (for 2×2 matrix A)	characteristic polynomial of A	Theorem 23.3.1(i)

Concept	**Meaning**	**Location in text**
eigenvector for $n \times n$ matrix A	*nonzero* $\mathbf{v} \in \mathbf{R}^n$ for which $A\mathbf{v} = \lambda\mathbf{v}$ for some scalar λ	Definition 23.1.2
eigenvalue for $n \times n$ matrix A	scalar λ for which vector equation $A\mathbf{x} = \lambda\mathbf{x}$ has a *nonzero* solution $\mathbf{x} \in \mathbf{R}^n$	Definition 23.1.3
eigenline for $n \times n$ matrix A	line in $\mathbf{R}^n$ spanned by an eigenvector for A	Definition 23.1.3
λ-eigenspace for $n \times n$ matrix A	the null space $N(A - \lambda\mathrm{I}_n)$	end of Prop. 23.1.11
trace (of 2×2 matrix A)	sum of diagonal entries of A	Theorem 23.3.1
characteristic polynomial (of 2×2 matrix A)	polynomial $\lambda^2 - \mathrm{tr}(A)\lambda + \det(A)$	Theorem 23.3.1(i)

Result	**Meaning**	**Location in text**
for $n \times n$ matrix A, if $\mathbf{v}$ is an eigenvector with associated eigenvalue λ then so is every nonzero scalar multiple of $\mathbf{v}$	$A(c\mathbf{v}) = c(A\mathbf{v}) = c(\lambda\mathbf{v}) = \lambda(c\mathbf{v})$ for $c \in \mathbf{R}$	(23.1.1)
for $\theta \neq 0^\circ, 180^\circ$, A_θ has no eigenvector	rotation moves line through $\mathbf{0}$ to *another* such line, so has no eigenvector	Example 23.1.7
determinant and null space description for eigenvalues and eigenvectors	for $n \times n$ matrix A: λ is eigenvalue precisely when $\det(A - \lambda\mathrm{I}_n) = 0$ (only used with $n = 2$), and then the eigenvectors of A with eigenvalue λ are the nonzero vectors in $N(A - \lambda\mathrm{I}_n)$	Prop. 23.1.11, Rem. 23.1.13
standard basis vectors are eigenvectors for diagonal matrices	diagonal $n \times n$ matrix D has $\mathbf{e}_i$ as eigenvector with eigenvalue d_{ii}	Example 23.2.1
for 2×2 matrix A, eigenvalues are roots in $\mathbf{R}$ of characteristic polynomial	if P_A has roots $\lambda_1, \lambda_2 \in \mathbf{R}$ then these are the eigenvalues, with $\lambda_1 + \lambda_2 = \mathrm{tr}(A)$ and $\lambda_1\lambda_2 = \det(A)$	Thm. 23.3.1, Cor. 23.3.2

Skill	**Location in text**
visualize eigenvectors for A via lines through $\mathbf{0}$ on which the effect of T_A is multiplying by some scalar λ (the associated eigenvalue)	Figures 23.1.1, 23.1.2
for triangular $n \times n$ matrix A, read off eigenvalues from diagonal and find eigenvectors via back-substitution with triangular $A - \lambda\mathrm{I}_n$	Theorem 23.2.2, Example 23.2.3
for any 2×2 matrix A, compute its eigenvalues (if any exist) and find all eigenvectors for each	Examples 23.3.4, 23.3.5, 23.3.6

24. Applications of eigenvalues: Spectral Theorem, quadratic forms, and matrix powers

In Chapter 23 we learned about the new concepts of eigenvalue and eigenvector for a general $n \times n$ matrix, and saw both some worked examples with $n > 2$ as well as a general technique for computing these in the case $n = 2$. The importance of eigenvalues in any dimension throughout all of mathematics and quantitative fields is difficult to overstate (we listed many applications near the start of Chapter 23).

The goal of this chapter is to discuss at length a couple of fundamental applications that show up all over the place. The unifying theme of both applications is that each is a consequence of a remarkable result called the Spectral Theorem. In addition to giving a lot of worked numerical examples to illustrate these deeper aspects of eigenvalues, we discuss a wide array of contexts where the results are used. The theme of eigenvalues and their astonishing breadth of utility will be revisited again in Chapters 26 and 27.

By the end of this chapter, you should be able to:

- use eigenvalues and eigenvectors to analyze definiteness properties of symmetric matrices (and quadratic forms);
- use eigenvalues and eigenvectors to compute and draw qualitative conclusions about A^m for $n \times n$ matrices A and (possibly large) exponents m.

24.1. The Spectral Theorem. We saw in Theorem 5.2.2 that pairwise orthogonality for a collection of nonzero vectors in $\mathbf{R}^n$ implies linear independence. The notion of eigenvalue provides a new and rather different source of instances of linear independence (and orthogonality).

Theorem 24.1.1. Let A be an $n \times n$ matrix and $\mathbf{v}_1, \dots, \mathbf{v}_r$ a collection of eigenvectors for A with respective eigenvalues $\lambda_1, \dots, \lambda_r$, so $A\mathbf{v}_j = \lambda_j \mathbf{v}_j$ for all j.

(i) If the r eigenvalues are pairwise different then the collection of r vectors $\{\mathbf{v}_1, \dots, \mathbf{v}_r\}$ in $\mathbf{R}^n$ is linearly independent, so $r \leq n$. Thus, an $n \times n$ matrix cannot have more than n different eigenvalues.

(ii) If A is symmetric and $\lambda_i \neq \lambda_j$ then $\mathbf{v}_i \cdot \mathbf{v}_j = 0$. That is, for a *symmetric* $n \times n$ matrix eigenvectors for different eigenvalues are orthogonal to each other.

In Example 24.1.3 we will give an example of a symmetric 3×3 matrix with 3 different eigenvalues (and give a collection of corresponding eigenvectors, which will be pairwise orthogonal by inspection). Symmetry is essential: a breakdown of orthogonality in part (ii) for non-symmetric A is illustrated in Example 23.1.4. For those who are interested, a proof of Theorem 24.1.1 is given in Section 24.7.

Remark 24.1.2. We are *not* claiming that linearly independent eigenvectors have pairwise different eigenvalues, only that when the eigenvalues for a collection of eigenvectors are pairwise different then the eigenvectors are automatically linearly independent. This is addressed more fully in Section 24.3.

If you reflect on the initial definition of eigenvalue, it may seem surprising that an $n \times n$ matrix cannot have an uncontrolled number of eigenvalues (e.g., is it visually apparent that a 3×3 matrix cannot have more than 3 different eigenvalues?). In Math 53, Math 104, and Math 113 the concept of "characteristic polynomial" is explored and gives an algebraic way to understand why an $n \times n$ matrix cannot have more than n different eigenvalues; this circle of ideas is introduced near the end of Appendix E (see Theorem E.5.1 and Remark E.5.2).

The concept of eigenvector makes sense for any $n \times n$ matrix A, but the main case for applications in this book is when A is *symmetric* (i.e., $A = A^\top$). In that case something special happens, which we first illustrate with an example.

Example 24.1.3. For $A = \begin{bmatrix} 16 & -2 & -6 \\ -2 & 19 & -3 \\ -6 & -3 & 27 \end{bmatrix}$, we saw in Example 23.1.1 that $\mathbf{v} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$ is an eigenvector of A with eigenvalue 12. You can also verify (please do) that $\mathbf{v}' = \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix}$ and $\mathbf{v}'' = \begin{bmatrix} -2 \\ -1 \\ 5 \end{bmatrix}$ are eigenvectors of A with respective eigenvalues 20 and 30 (i.e., $A\mathbf{v}' = 20\mathbf{v}'$ and $A\mathbf{v}'' = 30\mathbf{v}''$), and that $\{\mathbf{v}, \mathbf{v}', \mathbf{v}''\}$ is a collection of nonzero *orthogonal* 3-vectors (so this triple is an *orthogonal basis* of $\mathbf{R}^3$). ■

Example 24.1.3 is a special case of the following fundamental fact due to Cauchy[22] (see [Ha1, pp. 563-565], [Ha3, Sec. 4.4], and [Steen] for interesting historical context); it is proved in Section B.3.

Theorem 24.1.4 (Spectral Theorem). Let A be a *symmetric* $n \times n$ matrix. There is an orthogonal basis $\mathbf{w}_1, \dots, \mathbf{w}_n$ of $\mathbf{R}^n$ consisting of eigenvectors for A. The corresponding eigenvalues are all of the eigenvalues for A: if $\mathbf{w}_j$ has eigenvalue λ_j then any eigenvalue of A equals some λ_j.

In visual terms, $T_A : \mathbf{R}^n \to \mathbf{R}^n$ has the effect of λ_j-multiplication in the direction of $\mathbf{w}_j$ for all j.

The collection of eigenvalues of a square matrix is called its *spectrum*, so a theorem giving information about the eigenvalues of a special class of matrices is often called a "spectral theorem". The relationship with the word "spectrum" in physics is discussed in Example 24.6.1, and the central role of the Spectral Theorem throughout modern data science is discussed in Section 27.3.

Example 24.1.5. (i) Diagonal matrices are symmetric, and in Example 23.2.1 we saw that the standard basis for $\mathbf{R}^n$ consists of eigenvectors for any such matrix (the eigenvalue for the ith standard basis vector is the ith diagonal entry of such a matrix).

(ii) If V is a subspace of $\mathbf{R}^n$, and P is the $n \times n$ matrix of $\mathbf{Proj}_V : \mathbf{R}^n \to \mathbf{R}^n$ then P is symmetric (Proposition 20.3.10). Any nonzero $\mathbf{v} \in V$ satisfies $P\mathbf{v} = \mathbf{v} = 1 \cdot \mathbf{v}$, so $\mathbf{v}$ is an eigenvector of P with eigenvalue 1. Likewise, any nonzero $\mathbf{v}' \in V^\perp$ satisfies $P\mathbf{v}' = \mathbf{0} = 0 \cdot \mathbf{v}'$ (see Example 21.3.3), so $\mathbf{v}'$ is an eigenvector of P with eigenvalue 0. ■

Remark 24.1.6. Only for *symmetric* A are we claiming to have an abundant supply of pairwise orthogonal eigenvectors! A non-symmetric $n \times n$ matrix A may not have any eigenvectors in $\mathbf{R}^n$ (essentially due to the fact that polynomials with coefficients in $\mathbf{R}$ can fail to have any roots in $\mathbf{R}$); we saw this in Example 23.1.7 for rotations of $\mathbf{R}^2$ around the origin through any angle other than 0° and 180°.

Even if there is a basis of eigenvectors in $\mathbf{R}^n$ with distinct eigenvalues, beyond the symmetric case these eigenvectors are *not* all pairwise orthogonal. For instance, in Example 23.1.4 we gave a pair of eigenvectors $\{\mathbf{v}, \mathbf{v}'\}$ for a specific 2×2 non-symmetric matrix, with corresponding eigenvalues 7 and -5. By inspection the dot product of those two vectors is $7 \neq 0$, so those two eigenvectors are non-orthogonal.

The Spectral Theorem assures us that (in any dimension) such unfortunate situations never occur for symmetric matrices; this is one reason that the Spectral Theorem is so remarkable.

Given an orthogonal basis $\{\mathbf{w}_i\}$ of eigenvectors of a symmetric $n \times n$ matrix A, the Fourier formula

$$\mathbf{v} = \sum_{i=1}^{n} \left(\frac{\mathbf{v} \cdot \mathbf{w}_i}{\mathbf{w}_i \cdot \mathbf{w}_i} \right) \mathbf{w}_i \tag{24.1.1}$$

---

[22] Augustin-Louis Cauchy (1789-1857) was a prolific and influential French mathematician who worked on differential equations and mathematical physics [Tr], and made pioneering contributions to the modern formulation of calculus and linear algebra (e.g., he was the first to define $n \times n$ determinants [Ha3, Sec. 4.3]). His collected works occupy 27 volumes!

> from Theorem 5.3.6 that *expresses any vector* $\mathbf{v}$ *in terms of orthogonal eigenvectors* will be very useful. In this chapter we use it to solve problems such as: (i) determining when $q_A(\mathbf{v}) = \mathbf{v}^\top A\mathbf{v}$ is positive or negative (a very significant application because in Chapter 26 it **will help us to understand (in terms of eigenvalues) how functions behave near a critical point**, and (ii) describing the behavior of $A^m\mathbf{v}$ for big m. (The optional Section 24.6 gives many scientific applications of the Spectral Theorem.) We repeat the slogan for emphasis: *write everything in terms of* (orthogonal) *eigenvectors*.

24.2. Eigenvectors and quadratic forms. Consider the function $q_A(\mathbf{v}) = \mathbf{v}^\top A\mathbf{v}$ for a symmetric $n \times n$ matrix A When $n = 1$, so $A = [a]$ for some nonzero $a \in \mathbf{R}$, the graph in $\mathbf{R}^2$ of $q_A(x) = ax^2$ is a parabola pointing up or down (for $a > 0$ and $a < 0$ respectively); in the "degenerate" case $a = 0$ it is the x-axis $y = 0$ (a horizontal line). For $n = 2$ the graph $z = q_A(x, y)$ in $\mathbf{R}^3$ can be a "paraboloid" (cone-like with a rounded-off tip) pointing up or down as on the left in Figure 24.2.1 below, or saddle-shaped such as for $z = x^2 - y^2$ in Figure 10.2.3 (with associated contour plot in Figure 10.2.2), or a surface swept out by parallel copies of a single parabola (e.g., $z = 3x^2$) or a horizontal plane (e.g., $z = 0$).

There is more to say for $n = 2$: the horizontal slices of the paraboloid can be general ellipses (rather than circles), typically with axes of symmetry that are *not* the coordinate axes. Figure 24.2.1 is a typical "paraboloid" example, with horizontal slices that are ellipses tilted relative to the coordinate axes.

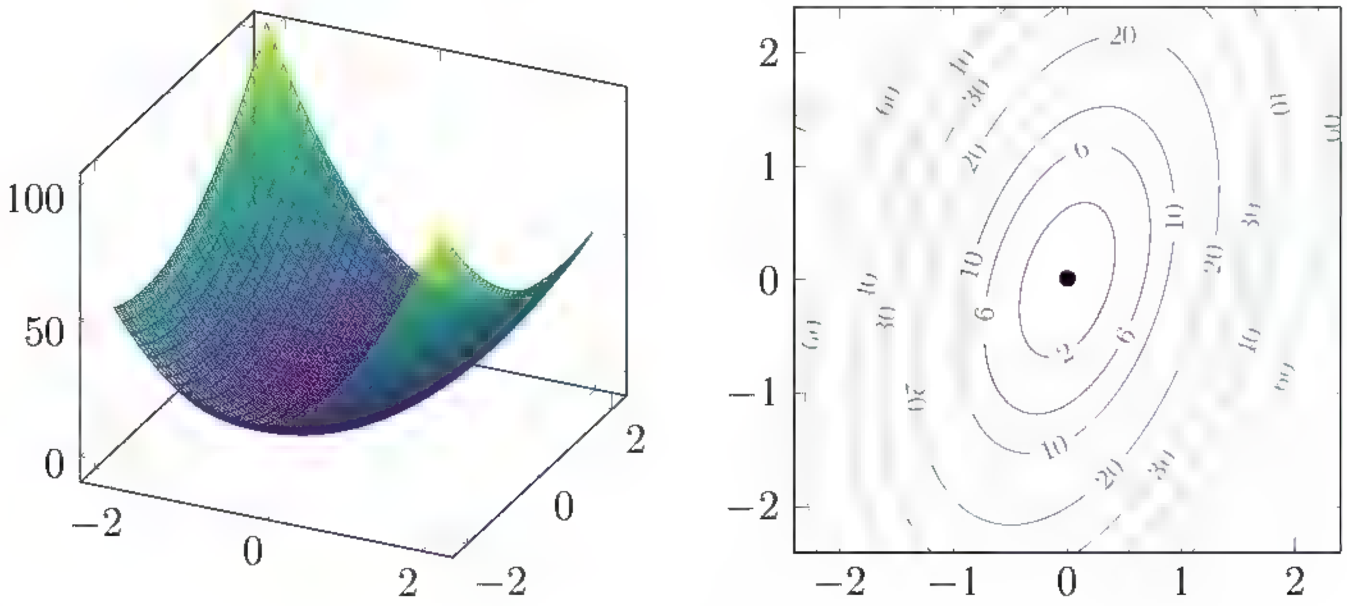


FIGURE 24.2.1. The paraboloid graph and contour plot for $q(x, y) = 13x^2 - 6xy + 5y^2$

In Definition 20.3.1 we introduced *quadratic forms*, and the dictionary between n-variable quadratic forms and symmetric $n \times n$ matrices was illustrated in Examples 20.3.11–20.3.12. The basic question, which we want to tackle for *any* n (pictures are available only for $n \le 3$), is this: how can we understand the geometry of level sets of an n-variable quadratic form in terms of its coefficients? This will lie at the heart of the *multivariable second derivative test* in Chapter 26, and it will be answered via the Spectral Theorem applied to the corresponding symmetric matrix!

Example 24.2.1. Let's now recover the geometry of the contour plot of $q(x, y) = 13x^2 - 6xy + 5y^2$ in Figure 24.2.1 by working with eigenvalues and eigenvectors for the symmetric matrix $A = \begin{bmatrix} 13 & -3 \\ -3 & 5 \end{bmatrix}$ associated with q. (In (24.2.2) the technique will be adapted to work *in general* for any n.)

By Theorem 23.3.1, A has eigenvalues $\lambda_1 = 14$ and $\lambda_2 = 4$. Solving $A\mathbf{x} = \lambda_i\mathbf{x}$ yields respective eigenvectors $\mathbf{w}_1 = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$ and $\mathbf{w}_2 = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$ that are orthogonal (such orthogonality is a general feature of symmetric matrices). This will be done in detail in Example 25.4.3, where we will also see that for the associated *unit* eigenvectors $\mathbf{w}_1' = \mathbf{w}_1/\|\mathbf{w}_1\| = \mathbf{w}_1/\sqrt{10}$ and $\mathbf{w}_2' = \mathbf{w}_2/\|\mathbf{w}_2\| = \mathbf{w}_2/\sqrt{10}$, if a general $\mathbf{v} = x\mathbf{e}_1 + y\mathbf{e}_2 \in \mathbf{R}^2$ is written in terms of the *orthonormal basis* $\{\mathbf{w}_1', \mathbf{w}_2'\}$ as $\mathbf{v} = t_1'\mathbf{w}_1' + t_2'\mathbf{w}_2'$ then

$$q(x,y) = q_A(\mathbf{v}) = q_A(t_1'\mathbf{w}_1' + t_2'\mathbf{w}_2') = \lambda_1 {t_1'}^2 + \lambda_2 {t_2'}^2 = 14{t_1'}^2 + 4{t_2'}^2.$$

(Explicitly, $t_j' = \mathbf{v}\cdot\mathbf{w}_j'$: we have $\mathbf{v}\cdot\mathbf{w}_1' = (t_1'\mathbf{w}_1' + t_2'\mathbf{w}_2')\cdot\mathbf{w}_1' = t_1'(\mathbf{w}_1'\cdot\mathbf{w}_1') + t_2'(\mathbf{w}_2'\cdot\mathbf{w}_1') = t_1'(1) + t_2'(0) = t_1'$ and similarly $\mathbf{v}\cdot\mathbf{w}_2' = t_2'$.)

The miracle here is that when $\mathbf{v}$ is *expressed in terms of an orthogonal basis of eigenvectors* for the symmetric matrix A, the value $q(\mathbf{v}) = q_A(\mathbf{v})$ at $\mathbf{v}$ of the associated quadratic form $q = q_A$ is given by an expression with *no cross-terms* (i.e., there is no $t_1't_2'$-term)! This will be explained in general in (24.2.2); it is saying that if we *rotate our orthonormal reference frame* from the standard basis $\{\mathbf{e}_1, \mathbf{e}_2\}$ to the basis $\{\mathbf{w}_1', \mathbf{w}_2'\}$ then the expression for q becomes the "$ax_1^2 + bx_2^2$" type whose elliptical shape we can see directly from the positivity of the coefficients 14 and 4 that are the eigenvalues of the symmetric matrix.

Superimposing $\{\pm\mathbf{w}_1', \pm\mathbf{w}_2'\}$ onto the contour plot from Figure 24.2.1 yields Figure 24.2.2 below, exhibiting visually that the perpendicular lines spanned by these unit eigenvectors are the lines of symmetry for those level curves $q = c$. The geometric property of being lines of symmetry corresponds to the algebraic property that in this tilted reference frame the equation for each level curve has *no cross-terms*.

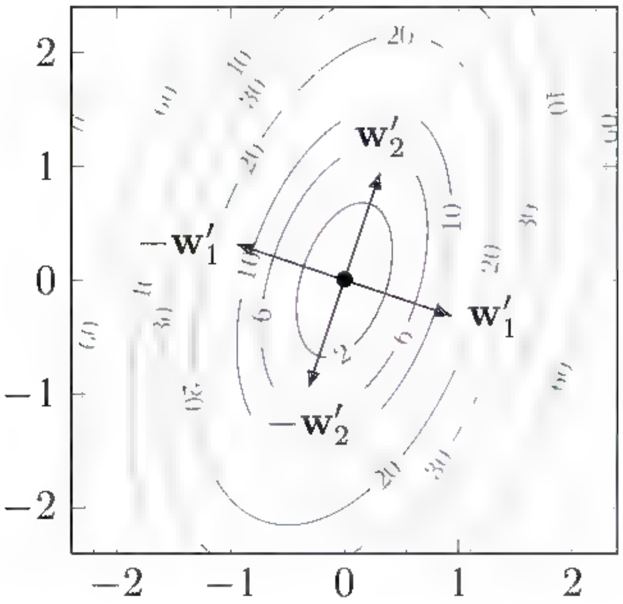


FIGURE 24.2.2. $q_A(x\mathbf{e}_1 + y\mathbf{e}_2) = 13x^2 - 6xy + 5y^2$ becomes $14{t_1'}^2 + 4{t_2'}^2$ when $x\mathbf{e}_1 + y\mathbf{e}_2$ is expressed in terms of the orthonormal basis $\{\mathbf{w}_1', \mathbf{w}_2'\}$ of eigenvectors for A.

The general lesson will be both algebraic and geometric: if we write everything in terms of an orthonormal basis of eigenvectors for a symmetric matrix then (i) the algebraic description of the associated quadratic form simplifies a lot (no cross-terms!), and (ii) the lines spanned by such eigenvectors are "lines of symmetry" of the level sets of the associated quadratic form. ■

We now *define* a notion of "positivity" for general symmetric matrices and quadratic forms:

Definition 24.2.2. For an $n \times n$ symmetric matrix A and the associated quadratic form $q_A(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x} = \mathbf{x}\cdot(A\mathbf{x})$, we say A and q_A are:

- *positive-definite* if $q_A(\mathbf{v}) > 0$ for all $\mathbf{v} \neq \mathbf{0}$, and *positive-semidefinite* if $q_A(\mathbf{v}) \geq 0$ for all $\mathbf{v} \neq \mathbf{0}$;
- *negative-definite* if $q_A(\mathbf{v}) < 0$ for all $\mathbf{v} \neq \mathbf{0}$, and *negative-semidefinite* if $q_A(\mathbf{v}) \leq 0$ for all $\mathbf{v} \neq \mathbf{0}$;
- *indefinite* if $q_A(\mathbf{v})$ takes both positive and negative values as $\mathbf{v}$ varies.

For emphasis: the phrases "A is positive-definite" and "q_A is positive-definite" mean *the same thing* (by definition!), and likewise for the other terms defined here.

Example 24.2.3. The matrices $A_+ = \begin{bmatrix} 2 & 0 \\ 0 & 5 \end{bmatrix}$ and $A_- = \begin{bmatrix} -3 & 0 \\ 0 & -4 \end{bmatrix}$ are respectively positive-definite and negative-definite since for $\mathbf{v} = \begin{bmatrix} x \\ y \end{bmatrix}$ we have $\mathbf{v}^\top A_+ \mathbf{v} = 2x^2 + 5y^2$ and $\mathbf{v}^\top A_- \mathbf{v} = -3x^2 - 4y^2$. ■

Example 24.2.4. The matrix $A = \begin{bmatrix} 1 & -3 \\ -3 & 9 \end{bmatrix}$ is positive-semidefinite but *not* positive-definite. Indeed,

$$\begin{bmatrix} x \\ y \end{bmatrix}^\top \begin{bmatrix} 1 & -3 \\ -3 & 9 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = x^2 - 6xy + 9y^2 = (x - 3y)^2$$

is always ≥ 0 (so A is positive-semidefinite) but it is not always > 0 when $\begin{bmatrix} x \\ y \end{bmatrix} \neq \mathbf{0}$. For example, $\mathbf{v}^\top A\mathbf{v} = 0$ when $\mathbf{v} = \begin{bmatrix} 3c \\ c \end{bmatrix}$ for any c. Likewise $-A$ is negative-semidefinite but *not* negative-definite. ■

Remark 24.2.5. Definition 24.2.2 really is a definition, not a theorem! For $n > 1$ there are *five* possibilities for nonzero quadratic forms q on $\mathbf{R}^n$: positive-definite, negative-definite, indefinite, and the two semidefinite (nonzero) options for q that are not definite. The point is that there are *many lines* $\ell = \mathrm{span}(\mathbf{v})$ in $\mathbf{R}^n$ through $\mathbf{0}$ when $n > 1$, with the values of q on ℓ away from $\mathbf{0}$ either all positive or all negative or or all 0 (since $q(c\mathbf{v}) = c^2 q(\mathbf{v})$ for $c \neq 0$) but which of the three cases occurs may *depend on* ℓ.

Our focus will be on positive-definite, negative-definite, and indefinite cases, but semidefinite cases that are *not* definite do arise in reality: [Sam, Ch. V; (61), (96)] is a negative-semidefinite example in consumer theory within economics, L in Remark 20.3.13 is positive-semidefinite but (see [MO, p. 1515]) can fail to be definite, and "kernel methods" in machine learning use semidefiniteness (see Section 24.5).

Remark 24.2.6. If a symmetric $n \times n$ matrix A is positive-definite or negative-definite then A is *invertible*. Indeed, to establish invertibility it is the same (by Theorem 18.3.3) to show for any n-vector $\mathbf{v} \neq \mathbf{0}$ that $A\mathbf{v} \neq \mathbf{0}$. Since $\mathbf{v}^\top A\mathbf{v} = q_A(\mathbf{v}) \neq 0$ (due to the "definiteness" of A because $\mathbf{v} \neq \mathbf{0}$), $A\mathbf{v} \neq \mathbf{0}$ as desired. In Remark 24.2.12 we will revisit this link to invertibility for symmetric matrices.

Example 24.2.7. In Example 20.3.14, we discussed that 3 different types of total energy in a vibrational mechanical system at a given time t are expressed as quadratic forms in terms of position or velocity vectors (at time t) for the parts of the system. This was done by means of specific symmetric matrices M, K, C: the mass matrix, the stiffness matrix, and the damping matrix. Since energy is always non-negative, it follows that each of these matrices is positive-semidefinite.

For an "unstable" structure (meaning that it can be displaced without applying any external forces) it can happen for special *nonzero* displacements $\mathbf{x}(t)$ from the rest position that the total elastic potential energy $(1/2)q_K(\mathbf{x}(t))$ is 0. When the structure is "stable" this cannot happen. On the other hand, even for a stable structure it can happen for special *nonzero* velocities $\mathbf{x}'(t)$ that the total dissipated energy $(1/2)q_C(\mathbf{x}'(t))$ is 0. So K and C are typically just positive-semidefinite, but in practice K is actually positive-definite. In such cases K is invertible (by Remark 24.2.6); the inverse matrix $A = K^{-1}$ is called the *flexibility matrix*, and it encodes how a unit force applied at one part of the system influences the displacement of other parts of the system.

The kinetic energy $(1/2)q_M(\mathbf{x}'(t))$ is *always* positive when $\mathbf{x}'(t) \neq \mathbf{0}$, because a system actually in motion always has some kinetic energy, so M is always positive-definite. This positive-definiteness is essentially obvious from the way M is defined in terms of the (positive) masses of the parts of the system, so the definiteness properties of K are more physically interesting than those of M. ■

When a symmetric $n \times n$ matrix A is not diagonal, so the quadratic form $q_A(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x}$ in n variables has "cross-terms" $x_i x_j$ with $i \neq j$, it is generally hopeless to determine just by staring at $q_A(\mathbf{x})$ if it is positive-definite, negative-definite, indefinite, or something else. Already in the case $n = 2$ this can be a puzzle: how about the two quadratic forms

$$10x_1^2 - 14x_1x_2 + 5x_2^2, \quad 8x_1^2 - 10x_1x_2 + 3x_2^2? \tag{24.2.1}$$

(The first is positive-definite and the second is indefinite; for a clean explanation see Example 26.3.4.)

This mystery is unlocked in a systematic way using eigenvalues. To explain this, recall that in (24.1.1) we gave a formula for how to write a general vector $\mathbf{v} \in \mathbf{R}^n$ in terms of an *orthogonal* basis of eigenvectors $\{\mathbf{w}_1, \dots, \mathbf{w}_n\}$ for a symmetric $n \times n$ matrix A, the existence of which is provided by the Spectral Theorem (Theorem 24.1.4)! We can use this to test for definiteness of the quadratic form $q_A(\mathbf{v}) = \mathbf{v}^\top A\mathbf{v} = \mathbf{v}\cdot(A\mathbf{v})$: if $\lambda_1, \dots, \lambda_n$ are the eigenvalues associated with the orthogonal eigenvectors $\mathbf{w}_1, \dots, \mathbf{w}_n$ then for $\mathbf{v} = \sum_{i=1}^n t_i\mathbf{w}_i$ orthogonality yields

$$\begin{aligned} q_A(\mathbf{v}) = \mathbf{v}\cdot(A\mathbf{v}) = \mathbf{v}\cdot\left(\sum_{i=1}^n t_iA\mathbf{w}_i\right) = \mathbf{v}\cdot\left(\sum_{i=1}^n t_i\lambda_i\mathbf{w}_i\right) &= \sum_{i=1}^n \lambda_i t_i(\mathbf{v}\cdot\mathbf{w}_i) \\ &= \sum_{i=1}^n \lambda_i t_i(t_i\mathbf{w}_i\cdot\mathbf{w}_i), \end{aligned}$$

(using $\mathbf{v}\cdot\mathbf{w}_i = (\sum_{j=1}^n t_j\mathbf{w}_j)\cdot\mathbf{w}_i = \sum_{j=1}^n((t_j\mathbf{w}_j)\cdot\mathbf{w}_i) = (t_i\mathbf{w}_i)\cdot\mathbf{w}_i$, the final equality since $(t_j\mathbf{w}_j)\cdot\mathbf{w}_i = t_j(\mathbf{w}_j\cdot\mathbf{w}_i) = 0$ for $j \neq i$ due to orthogonality of the $\mathbf{w}_k$'s). This yields the **diagonalization formula**

$$q_A(\mathbf{v}) = \sum_{i=1}^n \lambda_i(\mathbf{w}_i\cdot\mathbf{w}_i)t_i^2. \tag{24.2.2}$$

The cross-terms that plague a typical quadratic form have *completely disappeared* when we write everything in terms of the orthogonal basis $\mathbf{w}_1, \dots, \mathbf{w}_n$ of eigenvectors for A, with t_i^2 having coefficient $\lambda_i(\mathbf{w}_i\cdot\mathbf{w}_i)$ whose sign is the same as that of λ_i (when $\lambda_i \neq 0$) since $\mathbf{w}_i\cdot\mathbf{w}_i = \|\mathbf{w}_i\|^2 > 0$.

Example 24.2.8. The matrix $A = \begin{bmatrix} 16 & -2 & -6 \\ -2 & 19 & -3 \\ -6 & -3 & 27 \end{bmatrix}$ from Example 24.1.3 has eigenvalues $12, 20, 30$. The preceding calculations say that if we pick a nonzero vector $\mathbf{v}$ and (via (24.1.1)) write it as a linear combination $\mathbf{v} = t_1\mathbf{w}_1 + t_2\mathbf{w}_2 + t_3\mathbf{w}_3$ of respective orthogonal eigenvectors

$$\mathbf{w}_1 = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{w}_2 = \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix}, \quad \mathbf{w}_3 = \begin{bmatrix} -2 \\ -1 \\ 5 \end{bmatrix}$$

for those eigenvalues $12, 20, 30$ (with $\mathbf{w}_1\cdot\mathbf{w}_1 = 6$, $\mathbf{w}_2\cdot\mathbf{w}_2 = 5$, and $\mathbf{w}_3\cdot\mathbf{w}_3 = 30$) then

$$q_A(\mathbf{v}) = 12(\mathbf{w}_1\cdot\mathbf{w}_1)t_1^2 + 20(\mathbf{w}_2\cdot\mathbf{w}_2)t_2^2 + 30(\mathbf{w}_3\cdot\mathbf{w}_3)t_3^2 = 72t_1^2 + 100t_2^2 + 900t_3^2.$$

By inspection this is positive-definite: positivity of the coefficients is due to positivity of the eigenvalues!

We have not explained how to find eigenvectors and eigenvalues for the symmetric 3×3 matrix A (you will learn such things in a later math course, or see Section E.5), but once you have them in hand (e.g., if $n = 2$, or if we give them to you as above) such definiteness is something you can check. ■

Example 24.2.9. For $A = \begin{bmatrix} 1 & 2 \\ 2 & -2 \end{bmatrix}$, we computed in Example 23.3.5 (based on Theorem 23.3.1) that $\mathbf{w}_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ is an eigenvector with eigenvalue $\lambda_1 = 2$ and $\mathbf{w}_2 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$ is an eigenvector of A with eigenvalue $\lambda_2 = -3$. (If the $\mathbf{w}_i$'s had been handed to us, as would happen beyond the 2×2 case, we can also verify these properties directly.) Note that $\mathbf{w}_1 \cdot \mathbf{w}_2 = 0$ (this is related to A being symmetric), so $\{\mathbf{w}_1, \mathbf{w}_2\}$ is an orthogonal basis of $\mathbf{R}^2$ consisting of eigenvectors of A.

Determining the definiteness nature of $q_A\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} x \\ y \end{bmatrix}^\top \begin{bmatrix} 1 & 2 \\ 2 & -2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = x^2 + 4xy - 2y^2$ (written in standard coordinates) is not so apparent. But when q_A written in terms of the orthogonal basis $\{\mathbf{w}_1, \mathbf{w}_2\}$ of eigenvectors for A then the definiteness aspect is seen: if a vector $\mathbf{v}$ is written as $t_1\mathbf{w}_1 + t_2\mathbf{w}_2$ then

$$\begin{aligned} q_A(\mathbf{v}) = (t_1\mathbf{w}_1 + t_2\mathbf{w}_2) \cdot A(t_1\mathbf{w}_1 + t_2\mathbf{w}_2) &= (t_1\mathbf{w}_1 + t_2\mathbf{w}_2) \cdot (t_1 A\mathbf{w}_1 + t_2 A\mathbf{w}_2) \\ &= (t_1\mathbf{w}_1 + t_2\mathbf{w}_2) \cdot (2t_1\mathbf{w}_1 - 3t_2\mathbf{w}_2) \\ &= 2(\mathbf{w}_1 \cdot \mathbf{w}_1)t_1^2 - (\mathbf{w}_1 \cdot \mathbf{w}_2)t_1 t_2 - 3(\mathbf{w}_2 \cdot \mathbf{w}_2)t_2^2 \\ &= 2(\mathbf{w}_1 \cdot \mathbf{w}_1)t_1^2 - 3(\mathbf{w}_2 \cdot \mathbf{w}_2)t_2^2 \\ &= 10t_1^2 - 15t_2^2, \end{aligned}$$

where the second to last equality uses that $\mathbf{w}_1 \cdot \mathbf{w}_2 = 0$ (and is an instance of (24.2.2)) and the final equality uses that $\mathbf{w}_1 \cdot \mathbf{w}_1 = 5$ and $\mathbf{w}_2 \cdot \mathbf{w}_2 = 5$. Since the coefficients 10 and -15 have opposite signs, by varying t_1 and t_2 (e.g., setting one of them to be zero and the other to be nonzero) we see that $q_A(\mathbf{v})$ is sometimes positive and sometimes negative. The indefiniteness for q_A is due to λ_1 and λ_2 having opposite signs, and similarly to Example 24.2.1 the level curves $q_A(x, y) = c$ for nonzero c are hyperbolas (aligned with the lines through $\pm\mathbf{w}_1$ and $\pm\mathbf{w}_2$). ■

Returning to (24.2.2) in general, since $\mathbf{w}_i \cdot \mathbf{w}_i = \|\mathbf{w}_i\|^2 > 0$ and $t_i^2 \geq 0$ for all i, with $t_i^2 > 0$ for some i when $\mathbf{v} \neq \mathbf{0}$, we obtain the following result, resting on the Spectral Theorem to relate eigenvalues to Definition 24.2.2. (Some references present the result below as a definition, but then the relation with Definition 24.2.2 is recast as a hard theorem – real effort cannot disappear by manipulating definitions.)

Proposition 24.2.10. In the terminology of Definition 24.2.2, a symmetric $n \times n$ matrix A is:

(i) positive-definite when its eigenvalues are all positive;
(ii) negative-definite when its eigenvalues are all negative;
(iii) indefinite when some eigenvalue is positive and some eigenvalue is negative;
(iv) positive-semidefinite but not positive-definite when all eigenvalues are ≥ 0 and 0 is an eigenvalue; similarly for negative-semidefinite but not negative-definite using "≤ 0" instead of "≥ 0".

WARNING. A symmetric matrix with both positive and negative matrix entries need *not* be indefinite! For example, $A = \begin{bmatrix} 13 & -3 \\ -3 & 5 \end{bmatrix}$ has characteristic polynomial $\lambda^2 - 18\lambda + 56$ with roots $4, 14 > 0$, so it is positive-definite. (Concretely, $13x^2 - 6xy + 5y^2 > 0$ for $(x, y) \neq (0, 0)$.) Likewise, a symmetric matrix with *all* entries positive need *not* be positive-definite! For example, $B = \begin{bmatrix} 11 & 20 \\ 20 & 2 \end{bmatrix}$ has characteristic polynomial $\lambda^2 - 13\lambda - 378$ with roots $27, -14$, so it is indefinite. (Concretely, $11x^2 + 40xy + 2y^2$ takes on both positive and negative values: evaluate it at $(1, 1)$ and $(1, -1)$.)

Remark 24.2.11. It is a **very serious error** to think that one can read off the definiteness properties of a symmetric matrix by staring at the signs of its entries (or of a quadratic form by staring at the signs of its coefficients). One must go deeper into the linear algebra, as in the WARNING above.

Remark 24.2.12. If a symmetric $n \times n$ matrix A is known to be positive-semidefinite then it is positive-definite *precisely when* it is invertible! One direction was already seen in Remark 24.2.6: positive-definite symmetric matrices are always invertible. But the reverse direction in the positive-semidefinite case lies much deeper because it rests on (24.2.2). Namely, suppose A is positive-semidefinite but *not* positive-definite. Then we want to show that A cannot ever be invertible in such cases.

The key point is to inspect (24.2.2) in the positive-semidefinite case: by inspecting the right side, the non-negativity for all inputs $\mathbf{v}$ amounts to all eigenvalues λ_i being ≥ 0, yet the (assumed) failure of positive-definiteness tells us that we can't have all $\lambda_i > 0$. Hence, since all $\lambda_i \geq 0$, necessarily some λ_i vanishes. But then the corresponding eigenvector $\mathbf{w}_i$ is a *nonzero* vector satisfying $A\mathbf{w}_i = \lambda_i \mathbf{w}_i = 0\mathbf{w}_i = \mathbf{0}$. For an invertible $n \times n$ matrix M the vector equation $M\mathbf{x} = \mathbf{0}$ has $\mathbf{x} = \mathbf{0}$ as its *only* solution, so we conclude that A really *cannot* be invertible, as desired.

24.3. Further observations on eigenvalues. Let A be an $n \times n$ matrix (possibly not symmetric). There can be *lots* of eigenvectors of A with a given eigenvalue λ (whereas in contrast if we are given the eigenvector then there is only one eigenvalue for it, as we noted in Remark 23.1.10). This is just expressing the fact that the null space $N(A - \lambda \mathrm{I}_n)$, which is a linear subspace of $\mathbf{R}^n$ that is nonzero since λ is an eigenvalue, might have dimension larger than 1. To illustrate this in the case $n = 3$, consider the matrix $A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$. The orthogonal (so linearly independent) vectors $\mathbf{v} = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$, $\mathbf{v}' = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}$, $\mathbf{v}'' = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ are eigenvectors with respective eigenvalues $-1, -1, 2$ (as you can check directly: $A\mathbf{v} = -\mathbf{v}$, $A\mathbf{v}' = -\mathbf{v}'$, $A\mathbf{v}'' = 2\mathbf{v}''$). In particular, $\mathbf{v}$ and $\mathbf{v}'$ share the same eigenvalue $\lambda = -1$.

For the linear combination $\mathbf{w} = 3\mathbf{v} - 2\mathbf{v}' = \begin{bmatrix} -5 \\ 1 \\ 4 \end{bmatrix}$ we have

$$A\mathbf{w} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} -5 \\ 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 5 \\ -1 \\ -4 \end{bmatrix} = -\mathbf{w},$$

and the calculation works similarly for $\mathbf{w} = c\mathbf{v} + c'\mathbf{v}'$ for any scalars c and c':

$$A\mathbf{w} = A(c\mathbf{v} + c'\mathbf{v}') = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} -c + c' \\ c + c' \\ -2c' \end{bmatrix} = \begin{bmatrix} c - c' \\ -c - c' \\ 2c' \end{bmatrix} = -\mathbf{w}.$$

Any eigenvector $\mathbf{v}$ for an $n \times n$ matrix A is also an eigenvector for every power of A, using the corresponding powers of the eigenvalue λ: by feeding the relation $A\mathbf{v} = \lambda\mathbf{v}$ into itself repeatedly, we get $A^2\mathbf{v} = A(A\mathbf{v}) = A(\lambda\mathbf{v}) = \lambda(A\mathbf{v}) = \lambda(\lambda\mathbf{v}) = \lambda^2\mathbf{v}$ and similarly $A^3(\mathbf{v}) = A^2(A\mathbf{v}) = A^2(\lambda\mathbf{v}) = \lambda A^2(\mathbf{v}) = \lambda(\lambda^2\mathbf{v}) = \lambda^3\mathbf{v}$. Likewise in general:

$$\text{if } A\mathbf{v} = \lambda\mathbf{v} \text{ then } A^r\mathbf{v} = \lambda^r\mathbf{v} \text{ for any } r \geq 1. \tag{24.3.1}$$

Remark 24.3.1. If an $n \times n$ matrix A is *invertible* and λ is the eigenvalue of A for an eigenvector $\mathbf{v} \in \mathbf{R}^n$, then $\lambda \neq 0$ (since we saw in Example 23.1.9 that the occurrence of 0 as an eigenvalue happens precisely for non-invertible A, and we are assuming A is invertible).

Hence, it makes sense to ask if the formula $A^r\mathbf{v} = \lambda^r\mathbf{v}$ that we saw above for $r \geq 1$ also holds for $r = -1$; i.e., do we have $A^{-1}\mathbf{v} = \lambda^{-1}\mathbf{v}$? This is indeed true: multiplying both sides of the vector equation $A\mathbf{v} = \lambda\mathbf{v}$ against A^{-1} gives $A^{-1}(A\mathbf{v}) = A^{-1}(\lambda\mathbf{v}) = \lambda(A^{-1}\mathbf{v})$, yet the left side is $A^{-1}(A\mathbf{v}) = (A^{-1}A)\mathbf{v} = \mathrm{I}_n\mathbf{v} = \mathbf{v}$, so $\mathbf{v} = \lambda(A^{-1}\mathbf{v})$. Multiplying both sides by the scalar λ^{-1} gives $\lambda^{-1}\mathbf{v} = A^{-1}\mathbf{v}$.

Example 24.3.2. Let $A = \begin{bmatrix} 1 & 2 \\ 2 & -2 \end{bmatrix}$. We saw in Example 24.2.9 that $\mathbf{w}_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ and $\mathbf{w}_2 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$ constitute an orthogonal basis of $\mathbf{R}^2$ consisting of eigenvectors of A with respective eigenvalues $\lambda_1 = 2$ and $\lambda_2 = -3$.

Now let us see how we can analyze questions about A very efficiently using such an orthogonal basis. The basic idea is to *write everything in terms of the eigenvectors.*

(i) We can compute $A^5\mathbf{v}$ for any $\mathbf{v}$ by expressing $\mathbf{v}$ in terms of the eigenvectors. (In Chapter 16 we saw many contexts in which it is useful to be able to compute $A^m\mathbf{v}$ for big m.) Let's see this for $\mathbf{v} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$. First note that $A^5\mathbf{w}_1 = \lambda_1^5\mathbf{w}_1 = 32\mathbf{w}_1$ and $A^5\mathbf{w}_2 = \lambda_2^5\mathbf{w}_2 = -243\mathbf{w}_2$. Next, as in (24.1.1), we have

$$\begin{aligned}\mathbf{v} &= ((\mathbf{v}\cdot\mathbf{w}_1)/(\mathbf{w}_1\cdot\mathbf{w}_1))\mathbf{w}_1 + ((\mathbf{v}\cdot\mathbf{w}_2)/(\mathbf{w}_2\cdot\mathbf{w}_2))\mathbf{w}_2 \\ &= (11/5)\mathbf{w}_1 + (2/5)\mathbf{w}_2.\end{aligned}$$

Thus, $A^5\mathbf{v} = (11/5)A^5\mathbf{w}_1 + (2/5)A^5\mathbf{w}_2 = (11/5)(32\mathbf{w}_1) + (2/5)(-243\mathbf{w}_2) = \begin{bmatrix} 238 \\ -124 \end{bmatrix}$. Slick!

(ii) Let's solve the equation $A\mathbf{x} = \begin{bmatrix} 4 \\ 3 \end{bmatrix} = \mathbf{v}$. This is similar to the previous calculation, but now we'll use A^{-1} instead of A^5. As we saw in (i), $\mathbf{v} = (11/5)\mathbf{w}_1 + (2/5)\mathbf{w}_2$. Since the inverse matrix has the reciprocal eigenvalue against the same eigenvector (see Remark 24.3.1), we have $A^{-1}\mathbf{w}_1 = \lambda_1^{-1}\mathbf{w}_1 = \frac{1}{2}\mathbf{w}_1$ and $A^{-1}\mathbf{w}_2 = \lambda_2^{-1}\mathbf{w}_2 = -\frac{1}{3}\mathbf{w}_2$. Using the formulas as before, we therefore get

$$\mathbf{x} = A^{-1}\mathbf{v} = (11/5)A^{-1}\mathbf{w}_1 + (2/5)A^{-1}\mathbf{w}_2 = (11/5)(1/2)\mathbf{w}_1 + (2/5)(-1/3)\mathbf{w}_2 = \begin{bmatrix} 7/3 \\ 5/6 \end{bmatrix}.$$

$\blacksquare$

24.4. High powers of a matrix via eigenvalues. We saw in Chapter 16 that it is useful, when studying the evolution of a system over a long time, to be able to compute A^m for large integers m with A a fixed $n \times n$ matrix. It turns out that we can use eigenvalues and eigenvectors very effectively for this task. We will focus on the symmetric case, since then the Spectral Theorem provides a plentiful supply of eigenvalues. Everything we do in this section adapts to the non-symmetric case, which is needed for some topics from Chapter 16, but that involves complex numbers and so lies beyond the scope of this course.

As a warm-up illustration, we claim that for A as in Example 24.3.2 there is an exact formula

$$A^m = \begin{bmatrix} 1 & 2 \\ 2 & -2 \end{bmatrix}^m = \frac{1}{5}\begin{bmatrix} 4(2^m) + (-3)^m & 2(2^m - (-3)^m) \\ 2(2^m - (-3)^m) & 2^m + 4(-3)^m \end{bmatrix}, \tag{24.4.1}$$

where the significance of the numbers 2 and -3 being raised to powers is that they are the *eigenvalues* of A. The importance of this is that the *eigenvalue with biggest absolute value controls the behavior for large* m: if we pull out a factor of $(-3)^m$ everywhere we get

$$\begin{bmatrix} 1 & 2 \\ 2 & -2 \end{bmatrix}^m = \frac{(-3)^m}{5}\begin{bmatrix} 4(2/(-3))^m + 1 & 2\cdot((2/(-3))^m - 1) \\ 2((2/(-3))^m - 1) & (2/(-3))^m + 4 \end{bmatrix},$$

and $(2/(-3))^m \approx 0$ for large m since raising any number strictly between -1 and 1 (such as the *ratio of the eigenvalues* $2/(-3) = -2/3$) to very large powers rapidly approaches 0. Hence, replacing $(2/(-3))^m$ with 0 everywhere gives the clean approximation

$$\begin{bmatrix} 1 & 2 \\ 2 & -2 \end{bmatrix}^m \approx (-3)^m \begin{bmatrix} 1/5 & -2/5 \\ -2/5 & 4/5 \end{bmatrix} \tag{24.4.2}$$

for large m. The 2×2 matrix on the right is equal to

$$\frac{1}{\mathbf{w}\cdot\mathbf{w}}\mathbf{w}\mathbf{w}^\top, \tag{24.4.3}$$

where $\mathbf{w} = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$ is an eigenvector for the eigenvalue -3 with larger absolute value (you can check by direct calculation that (24.4.3) gives the matrix on the right side of (24.4.2), though it may seem like magic; Proposition 24.4.2 will show that such a formula is applicable very widely).

We repeat again for emphasis: the intervention of the eigenvalue 2 with smaller absolute value essentially disappears when we consider A^m for large m. This happens at the step where we observed that the ratio $2/(-3)$ of eigenvalues has absolute value less than 1 and thus its high powers are negligible. The general lesson, to be refined in Section 27.2, is this:

The behavior of A^m for large m is largely controlled by the eigenvalue(s) with largest absolute value and the associated eigenvector(s), called the "principal eigenvalue(s)" and "principal eigenvector(s)". (Striking applications of this insight to battle plans during World War II, also relevant to fighting genetic disease, are discussed here.)

A formula as in (24.4.1) is hopeless to discover by brute-force multiplication, so to find it (by a robust general method) we first give another way to think about the Spectral Theorem.

Theorem 24.4.1 (Interpretation of Spectral Theorem via a matrix decomposition). Let A be a symmetric $n \times n$ matrix with orthogonal eigenvectors $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n$, having corresponding eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$. Let W be the $n \times n$ matrix whose columns are the respective unit eigenvectors

$$\frac{\mathbf{w}_1}{\|\mathbf{w}_1\|}, \frac{\mathbf{w}_2}{\|\mathbf{w}_2\|}, \dots, \frac{\mathbf{w}_n}{\|\mathbf{w}_n\|}.$$

Then $W^\top = W^{-1}$ (i.e., W is an *orthogonal* matrix as discussed in Section 20.4), and

$$A = WDW^\top = WDW^{-1} \tag{24.4.4}$$

for the diagonal matrix $D = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}$ whose entries are the corresponding eigenvalues.

(The connection between (24.4.4) and the visualization at the end of Theorem 24.1.4 is given here.)

The formula (24.4.4) is explained at the end of Section 24.7 for those who are interested. As an illustration, we now apply it to $A = \begin{bmatrix} 1 & 2 \\ 2 & -2 \end{bmatrix}$ from (24.4.1). This matrix has the orthogonal eigenvectors $\mathbf{w}_1 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$ and $\mathbf{w}_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ with respective eigenvalues -3 and 2, with each $\mathbf{w}_i$ having length $\sqrt{5}$. Hence,

$$A = W \begin{bmatrix} -3 & 0 \\ 0 & 2 \end{bmatrix} W^\top = W \begin{bmatrix} -3 & 0 \\ 0 & 2 \end{bmatrix} W^{-1} \quad \text{for} \quad W = \begin{bmatrix} -1/\sqrt{5} & 2/\sqrt{5} \\ 2/\sqrt{5} & 1/\sqrt{5} \end{bmatrix} = (1/\sqrt{5}) \begin{bmatrix} -1 & 2 \\ 2 & 1 \end{bmatrix}.$$

(Multiply this out to check it really gives A.) Let us see what happens when we square A:

$$A^2 = WDW^{-1}WDW^{-1} = WD^2W^{-1},$$

where W^{-1} and W canceled on the inside! We can keep going with the same cancellation phenomenon:

$$A^3 = AAA = (WDW^{-1})(WDW^{-1})(WDW^{-1}) = WDDDW^{-1} = WD^3W^{-1}.$$

Likewise in general we have

$$A^m = WD^mW^{-1} = WD^mW^\top \text{ for any } m \geq 1. \tag{24.4.5}$$

This is useful because since D is diagonal, it is very easy to compute D^m.

For any $m \geq 1$ we have

$$\begin{aligned} A^m = WD^mW^\top &= \frac{1}{\sqrt{5}}\begin{bmatrix} -1 & 2 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} (-3)^m & 0 \\ 0 & 2^m \end{bmatrix}\frac{1}{\sqrt{5}}\begin{bmatrix} -1 & 2 \\ 2 & 1 \end{bmatrix} \\ &= \frac{1}{5}\begin{bmatrix} -1 & 2 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} -(-3)^m & 2(-3)^m \\ 2(2^m) & 2^m \end{bmatrix} \\ &= \frac{1}{5}\begin{bmatrix} 4(2^m)+(-3)^m & 2(2^m-(-3)^m) \\ 2(2^m-(-3)^m) & 2^m+4(-3)^m \end{bmatrix} \end{aligned}$$

(where the second to last equality combined both factors of $1/\sqrt{5}$ to get a single overall factor of $1/5$ in front). Therefore we have derived (24.4.1) using eigenvalues and eigenvectors.

Proposition 24.4.2. For a symmetric $n \times n$ matrix A (for any n), if there is an eigenvalue λ whose absolute value exceeds that of all other eigenvalues (this is called a *dominant eigenvalue*) and if the solutions to $A\mathbf{x} = \lambda\mathbf{x}$ constitute a line (as happens whenever there are n different eigenvalues) then for large m we have $A^m \approx (\lambda^m/(\mathbf{w}\cdot\mathbf{w}))\,\mathbf{w}\mathbf{w}^\top = \lambda^m\,\mathbf{Proj}_{\mathbf{w}}$ with $\mathbf{w}$ any eigenvector for A having eigenvalue λ.

This is the version of (24.4.2) that holds for any such A. The approximate stabilizing of A^m for large m is due to reasons closely related to the behavior of big powers of Markov matrices in Chapter 16 and that underlie PageRank in Appendix D (though Markov matrices are usually not symmetric).

Sometimes there is no dominant eigenvalue (e.g., the eigenvalues could be $5, 3, -5$), and when there is one then its eigenspace might have dimension > 1 (e.g., I_n has 1 as a dominant eigenvalue with eigenspace the entirety of $\mathbf{R}^n$). In such cases, Proposition 24.4.2 does not apply.

Example 24.4.3. To finish this section, we go beyond the symmetric case to illuminate Section 16.3. Although we stated Theorem 24.4.1 for symmetric matrices, the method of its proof (at the end of Section 24.7) applies more broadly. A motivating example is the non-symmetric 3×3 matrix

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 2 & 1 \end{bmatrix}$$

that arose in Section 16.3, where we saw that the sequence of numbers $a_1, a_2, a_3, \dots$ defined by the linear recurrence $a_m = a_{m-1} + 2a_{m-2} - a_{m-3}$ for $m \geq 4$ with initial terms $a_1 = a_2 = a_3 = 1$ satisfies

$$\begin{bmatrix} a_m \\ a_{m+1} \\ a_{m+2} \end{bmatrix} = A^{m-1}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \tag{24.4.6}$$

How can we get a handle on the powers A^{m-1} for big m? Once again, eigenvalues are the key.

Methods discussed in Section E.5 show that the matrix A has as its eigenvalues the three distinct roots of the cubic polynomial $x^3 - x^2 - 2x + 1$, given by

$$\lambda \approx 1.8109, \quad \lambda' \approx -1.2469, \quad \lambda'' \approx 0.4450.$$

The version of Theorem 24.4.1 that applies (with the same proof) to any $n \times n$ matrix M with distinct eigenvalues $\lambda_1, \dots, \lambda_n$ – no symmetry assumption on M – is that $M = WDW^{-1}$ where W and D are defined exactly as in Theorem 24.4.1 (but the matrix W is not orthogonal when M is not symmetric). Exactly as in our explanation of (24.4.1) above, we have $M^m = WD^mW^{-1}$ for any $m \geq 1$.

To apply this to our 3×3 matrix A, we have to compute an eigenvector for each of its 3 eigenvalues in order to determine the matrix "W" corresponding to A. Once this is done, feeding the resulting expression “$WD^{m-1}W^{-1}$” for A^{m-1} into the right side of (24.4.6) and doing some algebra (that we omit) yields an exact formula: $a_m = c\lambda^m + c'\lambda'^m + c''\lambda''^m$, where the coefficients c, c', c'' satisfy

$$\begin{bmatrix} 1 & 1 & 1 \\ \lambda & \lambda' & \lambda'' \\ \lambda^2 & \lambda'^2 & \lambda''^2 \end{bmatrix} \begin{bmatrix} c \\ c' \\ c'' \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}. \tag{24.4.7}$$

Since $|\lambda| > |\lambda'|, |\lambda''|$, for big m the term λ^m dominates in our exact formula for a_m and so $a_m \approx c\lambda^m$ for big m. By inverting the matrix on the left side of (24.4.7) one can obtain the exact formula $c = -(5/7)\lambda^2 + (3/7)\lambda + 12/7 \approx 0.1672$, so $a_m \approx c\lambda^m \approx (0.1672)(1.8109)^m$ for large m. ■

24.5. Application to support vector machines (optional). In Example 19.4.4 we introduced the concept of a “support vector machine” as an algorithm to build linear classifiers for multi-dimensional data: to find a “best” (affine) hyperplane H in $\mathbf{R}^n$ for separating n-vector data into two types, depending on training data $\mathbf{t}_1, \dots, \mathbf{t}_r \in \mathbf{R}^n$ (at least one of each type). That process relied upon multiple-constraint Lagrange multipliers, but the actual optimization problem that had to be solved in the end was stated without any explicit mention of the phrase “Lagrange multiplier”: for training data $\mathbf{t}_1, \dots, \mathbf{t}_r \in \mathbf{R}^n$ and signs $s_1, \dots, s_r = \pm 1$ (with both signs occurring), we want to maximize

$$\sum_{i=1}^{r} \lambda_i - \frac{1}{2} \sum_{i,i'=1}^{r} s_i s_{i'} (\mathbf{t}_i \cdot \mathbf{t}_{i'}) \lambda_i \lambda_{i'} \tag{24.5.1}$$

subject to the constraints $\lambda_1, \dots, \lambda_r \geq 0$ and $\sum_{i=1}^{r} \lambda_i s_i = 0$.

This method relies on knowing that the training data admits *some* separation by a hyperplane in $\mathbf{R}^n$, and in many situations there is no such separating hyperplane. Such failure is illustrated on the left in Figure 24.5.1, and the picture on the right in Figure 24.5.1 shows an important idea for how to circumvent that problem via a higher-dimensional space: find a (non-linear) function $f : \mathbf{R}^n \to \mathbf{R}^N$ with $N > n$ so that the modified data $f(\mathbf{t}_1), \dots, f(\mathbf{t}_r) \in \mathbf{R}^N$ admits a separating hyperplane (in $\mathbf{R}^N$).

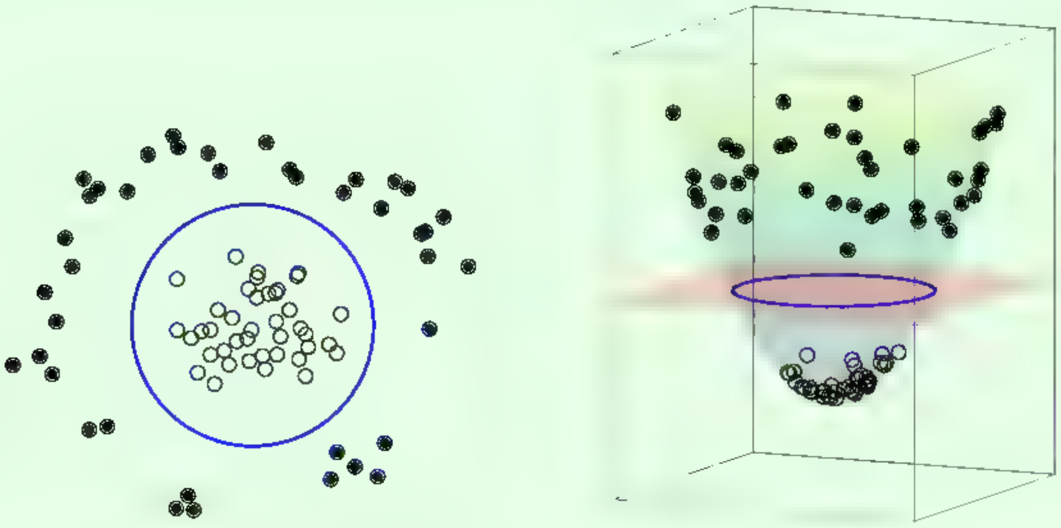


FIGURE 24.5.1. Data in $\mathbf{R}^n$ not separated by a hyperplane (as on the left) may admit a separating hyperplane after putting it into a higher-dimensional space by applying a suitable function $f : \mathbf{R}^n \to \mathbf{R}^N$ (whose output on the data is shown on the right).

But how do we find such an f? It may seem hopeless, but there is a way to essentially do this in practice thanks to the Spectral Theorem, as we now explain.

Suppose there were some (non-linear) $f : \mathbf{R}^n \to \mathbf{R}^N$ for which the data becomes amenable to the linear methods of Example 19.4.4 *after* applying f (set aside for a moment how such an f

would ever be found). When we apply the optimization task (24.5.1) to the modified training data $f(\mathbf{t}_1), \dots, f(\mathbf{t}_r)$, it becomes the task of maximizing

$$\sum_{i=1}^{r} \lambda_i - \frac{1}{2} \sum_{i,i'=1}^{r} s_i s_{i'} (f(\mathbf{t}_i) \cdot f(\mathbf{t}_{i'})) \lambda_i \lambda_{i'}$$

subject to the constraints $\lambda_1, \dots, \lambda_r \geq 0$ and $\sum_{i=1}^{r} \lambda_i s_i = 0$. For this task we don't need the full knowledge of f, but rather just the associated scalar-valued function $k_f(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) \cdot f(\mathbf{y})$ (for $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$). Moreover, once this task is solved, in practice the resulting non-linear classifier on new data $\mathbf{x} \in \mathbf{R}^n$ can be described in terms of the values $f(\mathbf{x}) \cdot f(\mathbf{t}_i) = k_f(\mathbf{x}, \mathbf{t}_i)$.

So what? If there were some way to *characterize* the scalar-valued $k(\mathbf{x}, \mathbf{y})$ which arise as $k_f(\mathbf{x}, \mathbf{y})$ for a (typically non-linear) function $f : \mathbf{R}^n \to \mathbf{R}^N$, then we could avoid having to build any f by instead working in the language of such functions k. That is, we may try a variety of such k's until we find one that works well for the task at hand: implicitly we would be applying some function f to the data but we never have to exhibit f explicitly. The ability to do exactly this is called the "kernel trick" (such a function k is called a "kernel" for reasons related to ideas in Math 175).

But the puzzle remains: how can we characterize the functions $k(\mathbf{x}, \mathbf{y})$ arising as k_f for (unknown) $f : \mathbf{R}^n \to \mathbf{R}^N$? Any such function k must satisfy two basic properties:

(i) (symmetry). It is symmetric in the sense that $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{y}, \mathbf{x})$ since $f(\mathbf{x}) \cdot f(\mathbf{y}) = f(\mathbf{y}) \cdot f(\mathbf{x})$.

(ii) (positive-semidefiniteness). For any $m \geq 1$ and collection of n-vectors $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbf{R}^n$, the $m \times m$ matrix $K = (k(\mathbf{x}_i, \mathbf{x}_j))$ is positive-semidefinite (i.e., $q_K(\mathbf{v}) = \mathbf{v}^\top K \mathbf{v} \geq 0$ for all $\mathbf{v} \in \mathbf{R}^m$). Indeed, if $k = k_f$ for some f then for the $m \times n$ matrix M with jth column $\mathbf{m}_j = f(\mathbf{x}_j)$ we have $K = (k_f(\mathbf{x}_i, \mathbf{x}_j)) = (f(\mathbf{x}_i) \cdot f(\mathbf{x}_j)) = (\mathbf{m}_i \cdot \mathbf{m}_j) = M^\top M$. This is a Gram matrix, hence positive-semidefinite ($q_{M^\top M}(\mathbf{v}) = \mathbf{v}^\top M^\top M \mathbf{v} = (M\mathbf{v})^\top (M\mathbf{v}) = (M\mathbf{v}) \cdot (M\mathbf{v}) = \|M\mathbf{v}\|^2 \geq 0$).

Sweeping a couple of technical issues under the rug, incredibly these two conditions on k are also *sufficient* to ensure $k = k_f$ for some f! This result is called *Mercer's Theorem*. (Examples of functions k satisfying conditions (i) and (ii) include the "radial kernel" $k(\mathbf{x}, \mathbf{y}) = e^{-c\|\mathbf{x}-\mathbf{y}\|^2}$ for $c > 0$ and the "polynomial kernel" $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + b)^d$ for $b \in \mathbf{R}$ and an integer $d > 0$.)

In Example 26.1.10 we will use the Spectral Theorem to prove a remarkable result of similar-sounding flavor: *every* symmetric positive-semidefinite matrix has the form $M^\top M$ for some matrix M. The proof of Mercer's Theorem uses an *infinite-dimensional* version of the Spectral Theorem (treated in Math 175), but controlling continuity properties of f lies deeper [Kii, Lemma 1].

24.6. The unreasonable effectiveness of mathematics (optional). In 1960, the Nobel prize-winning physicist Eugene Wigner published an essay [Wig] titled "The Unreasonable Effectiveness of Mathematics in the Natural Sciences", in which he wrote:

> *The miracle of the appropriateness of the language of mathematics for the formulation of the laws of physics is a wonderful gift which we neither understand nor deserve. We should be grateful for it and hope that it will remain valid in future research and that it will extend,* [. . .] *even though* [. . .] *to our bafflement, to wide branches of learning.*

Now that you have learned about the concept of eigenvectors and the Spectral Theorem, it seems apt to present some illustrations of what Wigner was talking about.

Example 24.6.1. In the early 20th century it was observed that electron motion within an atom causes the emission and absorption of energy at only specific discrete values (called its "spectrum"); this is one of the discoveries that led to quantum mechanics. In late 1925 Schrödinger wrote in a letter: "At the moment I am struggling with a new atomic theory. If only I knew more mathematics! I am very optimistic about this thing, and expect that, if only I can . . . solve it, it will be very beautiful."

Soon thereafter he made the key breakthrough in a 4-part series of papers called "Quantization as an Eigenvalue Problem" ([Sch1], [Sch2], [Sch3], [Sch4]), which showed that for each atom there is a type of (infinite-dimensional) matrix – later called the Schrödinger operator – that satisfies a version of the Spectral Theorem and its eigenvalues are *exactly* those discrete energies. Strictly speaking, Schrödinger used differential equations rather than matrices. But he showed in a specific case (in [Sch5]) that his quantum theory is equivalent to another due to Heisenberg based on matrices, and within a couple of years von Neumann established an infinite-dimensional Spectral Theorem and used it to prove a precise equivalence of the two quantum theories in general (see [Neu, Ch. I]).

This equality of atomic spectra with eigenvalues of a matrix in the mathematical model of atomic physics provided by quantum mechanics is an amazing historical coincidence for the following reason. The mathematician David Hilbert regarded the real-number eigenvalues of a symmetric matrix (as in the Spectral Theorem) as analogues of spectral lines showing the levels of emission and absorption of energy in atoms. For this reason, he referred to the collection of eigenvalues of any square matrix as its "spectrum", and he came up with the name "Spectral Theorem" accordingly. But this was all *before* the discovery of quantum mechanics! So, as illustrated by the quotation at the start of Part V, it was a huge surprise to Hilbert when Schrödinger later discovered that the spectrum of an atom actually is the collection of eigenvalues of a specific matrix (so what seemed like a mere analogy to Hilbert earlier was not just an analogy). ■

Example 24.6.2 (Principal Axes and the Tennis Racket Theorem). The Spectral Theorem in the case $n = 3$ has a striking physical consequence for the study of angular momentum in rotational mechanics, as we now explain. Imagine a rigid physical object spinning around in space under the influence of some forces (e.g., a spinning top, a rigid bar flying through the air, the Apollo 13 spacecraft, etc.), with the center of mass regarded as the origin. By physical reasoning, the dependence of the total angular momentum $\mathbf{L}$ on the total angular velocity $\vec{\omega}$ can be shown to respect vector operations on $\vec{\omega}$, so (by Theorem 14.2.1) $\mathbf{L}$ is equal to a matrix-vector product $\mathbf{I}\vec{\omega}$ for a 3×3 matrix $\mathbf{I}$ (sometimes called the *inertia tensor*, and generalizing the scalar formula $I\omega$ from introductory physics).

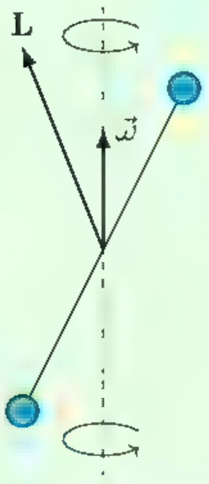


FIGURE 24.6.1. Angular momentum $\mathbf{L}$ and angular velocity $\vec{\omega}$ (shown for a spinning bar with equal masses at its ends) can lie on different lines through the center of mass.

The entries of $\mathbf{I}$ can be computed using integration formulas involving the geometry and mass distribution of the physical object being rotated. Explicit formulas for the entries of $\mathbf{I}$, or energy considerations in Lagrangian mechanics, show that $\mathbf{I}$ is *symmetric*. The fact that $\mathbf{L}$ is typically not pointing along the same line as $\vec{\omega}$ (in effect, $\mathbf{I}$ is not a scalar matrix) creates a lot of complexity in the study of rotational dynamics, especially when the physical object is irregularly-shaped (no axis of symmetry, etc.). Physical arguments show that the dot product $(1/2)(\vec{\omega} \cdot \mathbf{L})$ is equal to the rotational kinetic energy, which is positive whenever the angular velocity $\vec{\omega}$ is nonzero (since that corresponds to the presence of actual spinning). Substituting in the formula $\mathbf{L} = \mathbf{I}\vec{\omega}$, the rotational kinetic energy becomes $(1/2)(\vec{\omega} \cdot (\mathbf{I}\vec{\omega})) = (1/2)q_{\mathbf{I}}(\vec{\omega})$, so the symmetric matrix $\mathbf{I}$ is *positive-definite*.

Applying the Spectral Theorem to $\mathbf{I}$, if we suitably rotate our frame of reference by passing from an initial xyz-coordinate system to that of an orthonormal basis of eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ for $\mathbf{I}$, the effect of $\mathbf{I}$ in this new reference frame is "diagonal" with diagonal entries (the 3 eigenvalues of $\mathbf{I}$) that are positive. These eigenvalues are called the *principal moments of inertia* and they are intrinsic quantities, independent of any choice of coordinates (unlike entries in the matrix $\mathbf{I}$ in general).

The mutually orthogonal lines spanned by the $\mathbf{v}_i$'s are called the *principal axes* of the physical object, and their existence (via the Spectral Theorem, as we have seen) is called the *Principal Axis Theorem*. In case the eigenvalues of $\mathbf{I}$ are all different, so we can label them as $0 < \lambda_1 < \lambda_2 < \lambda_3$ from smallest to largest, there is a physical interpretation of the principal axes as follows. Let L_i be the line through the center of mass along the direction of an eigenvector for λ_i. If one imagines a rod running through the rigid body exactly along L_i and tries to spin the object in zero gravity exactly around that rod then the rod will remain perfectly in place, with no wobbling at all. This is surprising: for an irregularly-shaped body, it isn't apparent that any non-wobbling axis of spinning should exist!

There is an important caveat about this interpretation of the principal axes, because in the real world (rather than in physical idealizations in our head) no spinning around a line is perfectly exact. The principal axes L_3 and L_1 for the biggest and smallest eigenvalues are very *stable*: if one spins the rigid body approximately around either of those axes, any wobbling that occurs along it is tiny. However, spinning approximately around L_2 is *highly unstable*, causing unexpected behavior: instability for the middle eigenvalue is called the "Tennis Racket Theorem" and creates very counterintuitive motion for spinning objects in the absence of gravity called the "Dzhanibekov effect", shown in this short video. (If the rotating body is *not* rigid and so dissipates energy due to moving internal parts then by conservation of angular momentum even L_1-rotation is unstable: it turns into L_3-rotation because maximizing the moment of inertia minimizes the kinetic energy. The preference for L_3-rotation by dissipative bodies applies to the Earth but became widely recognized only after its effects on *Sputnik I* and *Explorer I* were observed; see [**Li**, App. A].)

For rotational dynamics of spinning motion (i.e., disregarding effects of gravity), such as in the "zero gravity" environment inside a spacecraft orbiting the earth, calculations become much easier when everything is described in terms of the principal axes. For example, the Euler equations of (rotational) motion for the angular velocity $\vec{\omega}$ become much more tractable to solve when written in the reference frame of the principal axes, and if one of the eigenvalues is larger than the others then the associated principal axis plays a dominant role in the rotational mechanics of the physical object.

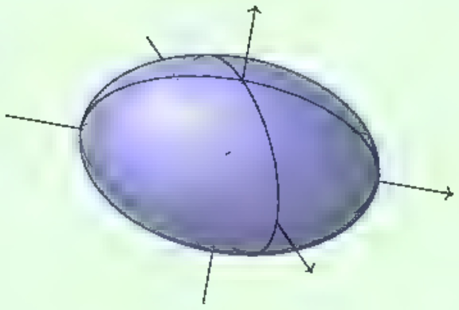


FIGURE 24.6.2. An ellipsoid and its three principal axes

For a rigid body with uniform density in the shape of an ellipsoid (this is essentially an egg-shape, but a bit more symmetric), the principal axes can be seen directly: as in Figure 24.6.2, they are certain mutually perpendicular lines through the center. The Principal Axis Theorem implies that for any rigid body B whatsoever, with an arbitrarily non-uniform density and irregular shape, its purely rotational dynamics behave *as if* it were a specific uniform-density solid ellipsoid E_B (with the same total mass and the same principal axes)! The eigenvalue for $\mathbf{I}$ along each principal axis for B determines the length of E_B in that direction, and the resulting volume of E_B and the mass of the rigid body B determine the uniform density of E_B. Usually E_B is called the *inertia ellipsoid* for B (see [**Feyn1**, Vol. II, Sec. 31.3-Sec. 31.4], which also discusses an "energy ellipsoid" for polarization of crystals).

To illustrate the utility of this miracle, first observe three geometric features of solid ellipsoids E:

(i) if the principal axes of E have different lengths then the only "rotational symmetry" (i.e., rotation around a line by less than a full rotation that carries E into itself) is 180-degree rotation around each of three perpendicular lines through the center of mass (namely, the principal axes);

(ii) if exactly 2 of the principal axes have the same length, so the plane P through those 2 axes makes a circular slice of E, the rotational symmetries of E are rotation by any angle around the line through the center of mass that is perpendicular to the plane P and 180-degree rotation around every line through the center of mass in that same plane;

(iii) if E has any rotational symmetry beyond what is accounted for in (i) and (ii) then all principal axes must have the same length and so E is a solid ball.

In particular, an ellipsoid with rotational symmetry that does not fit within cases (i) or (ii) must be in case (iii); i.e., it must be a solid ball (in which case of course it has the maximal possible rotational symmetry). Examples of such "extra rotational symmetry" include rotation by an angle different from 180° around two different lines through the center of mass and rotational symmetry around 3 different lines through the center of mass that are not all in the same plane and not mutually perpendicular.

Let's apply this to a non-spherical solid body B with uniform density that has either a rotational symmetry by other than 180° around two different lines through the center of mass or a rotational symmetry around 3 different lines through the center of mass which are not all in the same plane. For example, a solid cube C has 90° rotational symmetry around all 3 lines connecting centers of opposite faces, and 120° rotational symmetry around all 3 diagonal lines connecting opposite corners.

The inertia ellipsoid $E = E_B$ for such a B must inherit the same rotational symmetries (since E is designed to have the *same* rotational dynamics as B), so E must be a *solid ball* with uniform density. (If B is a cube with side length s, by physical arguments its spherical inertia ellipsoid can be shown to have radius $\sqrt{5/12}\,s$.) As a consequence, B's moment of inertia around any line through its center of mass (this is a measure of resistance to angular acceleration around that line, much as mass gives such a measure for linear motion due to Newton's force law) is *the same* as for any other such line since the equality of moments of inertia for all such lines holds for the solid ball E (due to the extensive rotational symmetry of E, which B does not have!). This conclusion for B is surprising, since many lines through the center of mass of B pass through B in an asymmetric manner; one wouldn't expect that the moment of inertia around such a line is the same for every line through the center. A variant of this "symmetry analysis" is given (for energy in crystals) near the end of [**Feyn1**, II, Sec. 31.3]. ■

Remark 24.6.3. The ability to diagonalize the symmetric matrix $\mathbf{I}$ via passage to a new *orthonormal reference frame* was originally discovered by Laplace in the 18th-century via arguments specific to the physical setting. The recognition of the role of symmetry of the matrix and the wider validity of the Spectral Theorem for symmetric $n \times n$ matrices for all n only came later, with work of Cauchy on celestial mechanics in 1829 (see [**Ha1**, pp. 563-565] and [**Ha3**, Sec. 4.4] for an historical survey).

Example 24.6.4. Let's return to the structural analysis of vibrational mechanical systems that we saw in Examples 20.3.14 and 24.2.7; you can keep in mind a collection of many springs linked together in a complicated way (a toy version of which is given in Example 10.1.3). There was a stiffness matrix K which turned out to be symmetric. We want to explain why that symmetry has useful consequences for the scientific analysis when one brings in the mathematics of the Spectral Theorem.

The definition of the stiffness matrix is as an $n \times n$ matrix where n is the number of "degrees of freedom" of the system; this n can be quite large (depending on the number of parts in the mechanical system). Informally, any possible vibration of the system can be described as a sum of contributions from vibrations along each degree of freedom. These degrees of freedom are something one sees with one's eyes by staring at the mechanical system and thinking about the ways in which different parts can vibrate. But when the equations of motion for the vibrations are written in terms of these degrees of freedom, they are a huge system of differential equations that is a gigantic mess.

Now comes the contribution of eigenvalues to escape the mess: the Spectral Theorem guarantees that there is an *orthogonal basis* of eigenvectors for the stiffness matrix. The eigenvectors can be interpreted as special types of vibration of the mechanical system; we'll call them "eigenvibrations". The corresponding eigenvalues are stress energies for these specific vibrations.

Why are the eigenvibrations useful, and not just some piece of theoretical math? Let's first clarify what it means that this collection of n eigenvibrations is (by its design) a *basis* for the "space of vibrations": instead of describing every possible vibration in terms of contributions from vibrations along each of the n degrees of freedom (this is something that can be physically seen), we can instead describe every vibration (uniquely) as a linear combination of the eigenvibrations. The same could be said for *any basis* of n vectors for the "space of vibrations", but the eigenvibrations are not just any old basis: they are (by design) *eigenvectors for the stiffness matrix* and are *orthogonal to each other*. By combining the orthogonality with the way the stiffness matrix appears in the equations of motion, it then follows from the eigenvector property that when everything is described as a linear combination of the eigenvibrations, the mathematics of the vibrational behavior of the original highly coupled system with n degrees of freedom becomes exactly the mathematics of a collection of n *unrelated* 1-dimensional spring systems! So one can apply tools developed for the 1-dimensional case, and reassemble the conclusions to learn information about the original highly coupled mechanical system.

In other words, the differential equations of motion that appear as a mess in terms of the physical notion of the n vibrational "degrees of freedom" dramatically simplify when everything is written in terms of the n eigenvibrations. The ability to reformulate the mathematics of the highly correlated physical systems in terms of the mathematics of a collection of unrelated 1-dimensional systems *has no physical interpretation at all*; one could say it is just an amazing mathematical trick (which relies on the application of a real theorem, namely the Spectral Theorem). Yet the process by which this is achieved provides the existence of n very special vibrations whose existence could never otherwise be detected, and those in turn are useful in a deeper study of the mechanics.

Unlike the very physical n "degrees of freedom" for vibrational motion, it is *absolutely impossible* to see the existence of the n eigenvibrations by staring at the original mechanical system. One really has to dig into the mathematics to see the presence of a huge *symmetric* matrix K and then apply the Spectral Theorem to it in order to deduce the existence of very special vibrations in terms of which the equations of motion take on a much more tractable form. So the Spectral Theorem, which may initially seem to be "just" pure mathematics, can be used to great practical effect in transforming messy equations of motion into a much more tractable form. In Section 27.1 we present the same technique in the more tangible context of a pair of linked pendulums (where we completely work out the passage to unrelated 1-dimensional systems). ■

Example 24.6.5 (Dinosaur tail eigenvalues). Data shows that for most mammals, the preferred walking speed and gait minimize the energy expenditure. Assuming this also holds for non-avian dinosaurs, a new insight in [BSS] was that the bouncing of the massive tail of the *Tyrannosaurus rex* should be incorporated into models for such energy minimization.

Under some assumptions concerning the musculature of the tail [BSS, Fig. S3], the bouncing is modeled by dividing the tail into 5 segments of equal length and treating it as a spring-mass system as illustrated in this video. The differential equations of motion for the tail are governed by a 10×10 matrix, and the numerical determination of its eigenvalues on a computer (combined with physiological information about the dinosaur) led to the surprising conclusion that the preferred *T. rex* walking speed was around 1.28 meters/sec, which is *slower* than the average human adult preferred walking speed of around 1.34-1.42 meters/sec! ■

Example 24.6.6 (Dead Sea Scrolls). A rather spectacular application of the Spectral Theorem arises in archaeology, or more broadly in the study of ancient manuscripts that are rolled up and too brittle to open. For example, the Dead Sea Scrolls are a series of biblical texts found on parchment in ancient caves near the Dead Sea beginning in the 1940's, and among the documents discovered from searches of the surrounding caves is the En-Gedi Scroll so badly burned (from long ago) that it is essentially a cylinder of charcoal.

In 2016, sophisticated computerized mathematical techniques based on differential geometry were applied to a 3-dimensional scan of this artifact to virtually "unwrap" it so that the original ink could be read! One of the key ingredients in the method rests on the following consequence of the Spectral Theorem in 3 dimensions (see [LMT, Section 3.3]). Let's call a positive semi-definite quadratic form $q : \mathbf{R}^3 \to \mathbf{R}$ *spherical* if $q(\mathbf{x}) = \|\mathbf{x}\|^2$, *planar* if $q(\mathbf{x}) = \|\mathbf{Proj}_{\mathcal{P}}(\mathbf{x})\|^2$ for some plane $\mathcal{P}$ through the origin (in effect, q depends on 2 variables: coordinates along P), and *stick-like* if $q(\mathbf{x}) = \|\mathbf{Proj}_L(\mathbf{x})\|^2$ for some line L through the origin (in effect, q depends on 1 variable: a coordinate alone L). The Spectral Theorem allows us to build up *any* positive semi-definite quadratic form $Q : \mathbf{R}^3 \to \mathbf{R}$ as a non-negative linear combination of these three special types of quadratic forms.

To see this, write $Q = q_A$ for a symmetric 3×3 matrix A. By the Spectral Theorem, there is an orthonormal basis of $\mathbf{R}^3$ consisting of A-eigenvectors, say $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$. By rearranging if necessary, we can assume the associated eigenvalues satisfy $\lambda_1 \geq \lambda_2 \geq \lambda_3$, and they are non-negative since Q is positive semi-definite. Define the plane $\mathcal{P} = \mathrm{span}(\mathbf{v}_1, \mathbf{v}_2)$ and line $L = \mathrm{span}(\mathbf{v}_1)$. Writing a 3-vector $\mathbf{x}$ as $\mathbf{x} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + a_3\mathbf{v}_3$, we have $\|\mathbf{x}\|^2 = \mathbf{x} \cdot \mathbf{x} = a_1^2 + a_2^2 + a_3^2$ (since the $\mathbf{v}_i$'s are an orthonormal basis of $\mathbf{R}^3$) and likewise $\|\mathbf{Proj}_{\mathcal{P}}(\mathbf{x})\|^2 = \|a_1\mathbf{v}_1 + a_2\mathbf{v}_2\|^2 = a_1^2 + a_2^2$ and $\|\mathbf{Proj}_L(\mathbf{x})\|^2 = \|a_1\mathbf{v}_1\|^2 = a_1^2$. Since the orthonormal $\mathbf{v}_i$'s satisfy $A\mathbf{v}_i = \lambda_i\mathbf{v}_i$, we have

$$\begin{aligned} Q(\mathbf{x}) = q_A(\mathbf{x}) &= \lambda_1 a_1^2 + \lambda_2 a_2^2 + \lambda_3 a_3^2 \\ &= \lambda_3(a_1^2 + a_2^2 + a_3^2) + (\lambda_2 - \lambda_3)(a_1^2 + a_2^2) + (\lambda_1 - \lambda_2)a_1^2 \\ &= \lambda_3\|\mathbf{x}\|^2 + (\lambda_2 - \lambda_3)\|\mathbf{Proj}_{\mathcal{P}}(\mathbf{x})\|^2 + (\lambda_1 - \lambda_2)\|\mathbf{Proj}_L(\mathbf{x})\|^2 \end{aligned}$$

with coefficients $\lambda_3, \lambda_2 - \lambda_3, \lambda_1 - \lambda_2$ that are all *non-negative.* ■

Example 24.6.7 (Deformation tensors and principal stresses). Another application of the Spectral Theorem arises in the context of deformations of solids, which we first considered in Example 13.5.7. There we had a solid body B in space undergoing a deformation $\mathbf{f} : B \to \mathbf{R}^3$, and we said that engineers refer to $(D\mathbf{f})(\mathbf{b})$ as the *deformation gradient*; actually, they often call it the *deformation gradient tensor*, where "tensor" is a word (derived from the Latin *tendere*, meaning "to stretch") that physicists and engineers use for a linear-algebra gadget – such as a vector or matrix or quadratic form

– expressed in a certain coordinate-dependent manner (mathematicians use the word with a related but more intrinsic meaning). The physical nature of the deformation process ensures (by the Chain Rule) that $(D\mathbf{f})(\mathbf{b})$ is invertible for all $\mathbf{b}$. Engineers often denote $(D\mathbf{f})(\mathbf{b})$ as $\mathbf{F}(\mathbf{b})$.

Many quantities of physical interest related to stress and strain in a deformation process are encoded in the deformation gradient tensor. To explain this, we require a result in linear algebra that (in a sense which is too much of a digression to explain here) generalizes polar coordinates on $\mathbf{R}^2$:

Theorem (Polar Decomposition). If A is an invertible $n \times n$ matrix then we can uniquely write $A = QS$ where Q is an orthogonal $n \times n$ matrix and S is a positive-definite symmetric $n \times n$ matrix. Moreover, also $A = S'Q$ for the same Q and a positive-definite symmetric $n \times n$ matrix S'.

For a proof, see Theorem B.4.2. As an illustration, $A = \begin{bmatrix} 3/5 & -8/5 \\ 4/5 & 6/5 \end{bmatrix}$ has Polar Decomposition

$$\begin{bmatrix} 3/5 & -4/5 \\ 4/5 & 3/5 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} = QS = A = S'Q = \begin{bmatrix} 41/25 & -12/25 \\ -12/25 & 34/25 \end{bmatrix} \begin{bmatrix} 3/5 & -4/5 \\ 4/5 & 3/5 \end{bmatrix};$$

in this case $S \neq S'$ though S and S' have the same eigenvalues $\lambda_1 = 1$ and $\lambda_2 = 2$ (with respective eigenvectors $\mathbf{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\mathbf{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and $\mathbf{v}_1' = \begin{bmatrix} 3/5 \\ 4/5 \end{bmatrix} = Q\mathbf{v}_1$, $\mathbf{v}_2' = \begin{bmatrix} -4/5 \\ 3/5 \end{bmatrix} = Q\mathbf{v}_2$).

The Polar Decomposition may look similar to the QR-decomposition, but it is deeper: the proof of the QR-decomposition only uses the Gram–Schmidt process, whereas the proof of the Polar Decomposition rests on the Spectral Theorem. Since $QS = A = S'Q$, so $S = Q^{-1}S'Q$, geometrically S' looks like S up to a change of reference frame defined by the rigid motion Q. In particular, the eigenvalues of S' are *the same* as those of S, though the orthonormal bases of eigenvectors for each (from the Spectral Theorem) generally do not agree: one such basis is carried to the other by applying the rigid motion Q (or Q^{-1}); the above 2×2 illustration exhibits these features.

Applying the Polar Decomposition with $n = 3$ to $(D\mathbf{f})(\mathbf{b})$, we obtain

$$Q(\mathbf{b})S(\mathbf{b}) = (D\mathbf{f})(\mathbf{b}) = S'(\mathbf{b})Q(\mathbf{b}) \tag{24.6.1}$$

for a 3×3 orthogonal matrix $Q(\mathbf{b})$ and positive-definite symmetric 3×3 matrices $S(\mathbf{b})$ and $S'(\mathbf{b})$. (In the materials science literature, $Q(\mathbf{b})$ is denoted as $R(\mathbf{b})$, $S(\mathbf{b})$ is denoted as $U(\mathbf{b})$, and $S'(\mathbf{b})$ is denoted as $V(\mathbf{b})$; we avoid that notation since in Chapter 22 we used R and U to denote upper triangular matrices, as is standard in applied linear algebra.) By the linear-approximation property $\mathbf{f}(\mathbf{b} + \mathbf{h}) \approx \mathbf{f}(\mathbf{b}) + ((D\mathbf{f})(\mathbf{b}))\mathbf{h}$ for small $\mathbf{h}$, the physical interpretation of (24.6.1) is that (up to a spatial translation by $\mathbf{f}(\mathbf{b})$) the orthogonal matrix $Q(\mathbf{b})$ encodes the part of the deformation process near $\mathbf{b}$ that is purely rotational (so no change in nearby distances) whereas $S'(\mathbf{b})$ and $S(\mathbf{b})$ record in different ways the part of the deformation process near $\mathbf{b}$ that involves a change in nearby distances. Engineers call $Q(\mathbf{b})$ the *rotation tensor*, $S(\mathbf{b})$ the *right stretch tensor*, and $S'(\mathbf{b})$ the *left stretch tensor*.

To explain the "stretch" terminology, since $S(\mathbf{b})$ and $S'(\mathbf{b})$ are symmetric they each have an orthogonal basis of eigenvectors by the Spectral Theorem and the eigenvalues of those eigenvectors are positive by Proposition 24.2.10(i). The orthogonal basis can be made into an orthonormal basis by scaling each eigenvector in the basis to have length 1, which doesn't change its eigenvalue. We have noted already that the eigenvalues for $S'(\mathbf{b})$ and $S(\mathbf{b})$ coincide (though the eigenvectors usually do not). These common eigenvalues are called the *principal stresses* at $\mathbf{b}$ because the physical interpretation of the expression $Q(\mathbf{b})S(\mathbf{b})$ for $(D\mathbf{f})(\mathbf{b})$ is that (up to a spatial translation) near $\mathbf{b}$ the deformation process consists of (i) first stretching or compressing along the lines spanned by the orthonormal basis of eigenvectors for $S(\mathbf{b})$ (called the *principal directions* for the deformation at $\mathbf{b}$) with the scaling factor along each eigenline (i.e., span of an eigenvector) given by the respective (positive) eigenvalue, and (ii) then applying the rotation $Q(\mathbf{b})$.

Likewise, the physical interpretation of the expression $S'(\mathbf{b})Q(\mathbf{b})$ for $(D\mathbf{f})(\mathbf{b})$ is that (up to a spatial translation) near $\mathbf{b}$ the deformation process consists of (i) first applying the rotation $Q(\mathbf{b})$, and (ii) then stretching or compressing along the lines spanned by the orthonormal basis of eigenvectors for $S'(\mathbf{b})$ with the scaling factor along each eigenline given by the respective (positive) eigenvalue. Preference among $Q(\mathbf{b})S(\mathbf{b})$ or $S'(\mathbf{b})Q(\mathbf{b})$ expresses preference among the reference frame of the external environment or of the body undergoing deformation.

In the context of the general Polar Decomposition we have

$$A^\top A = (QS)^\top(QS) = S^\top Q^\top QS = S(Q^\top Q)S = S^2$$

(using that $S^\top = S$ by symmetry of S, and $Q^\top Q = \mathrm{I}_n$ by orthogonality of Q), and similarly $AA^\top = (S'Q)(S'Q)^\top = S'QQ^\top S'^\top = S'(QQ^\top)S' = S'^2$. Thus, in the deformation setting we have

$$((D\mathbf{f})(\mathbf{b}))^\top(D\mathbf{f})(\mathbf{b}) = S(\mathbf{b})^2, \;\; (D\mathbf{f})(\mathbf{b})((D\mathbf{f})(\mathbf{b}))^\top = S'(\mathbf{b})^2.$$

In particular, the principal directions and the squares of the principal stresses at $\mathbf{b}$ are encoded in the (symmetric) Gram matrices $((D\mathbf{f})(\mathbf{b}))^\top(D\mathbf{f})(\mathbf{b})$ and $(D\mathbf{f})(\mathbf{b})((D\mathbf{f})(\mathbf{b}))^\top$ that are respectively called the *right Cauchy–Green tensor* and *left Cauchy–Green tensor*. These two matrices record the change in geometry near $\mathbf{b}$ disregarding rotational effects, up to squaring the principal stresses at $\mathbf{b}$. (The inverse of the left Cauchy–Green tensor arose in Cauchy's pioneering 1828 work on deformations of solids, and G. Green[23] introduced the right Cauchy–Green tensor 11 years later; neither Cauchy nor Green had the efficient language of modern linear algebra that we do.)

Observe how pervasive the Spectral Theorem and matrix algebra are in this discussion! The principal stresses and principal directions can be recorded in terms of analogues of the inertia ellipsoid from Example 24.6.2 called the *stress ellipsoid* and *strain ellipsoid* (which vary with $\mathbf{b} \in B$). Planes of symmetry through the centers of these ellipsoids are useful visual tools when discussing the fracturing of material under stress; also see [**Feyn1**, II, Sec. 31.6] (where the variation of the stress ellipsoid with $\mathbf{b} \in B$ is noted near the end). Materials scientists use the preceding concepts all over the place. ■

24.7. Some eigenvalue proofs. This section proves Theorem 24.1.1 and explains where the matrix decomposition (24.4.4) comes from. To prove Theorem 24.1.1, we use notation as in its statement.

PROOF. For part (i), to show that eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_r \in \mathbf{R}^n$ for an $n \times n$ matrix A are linearly independent when the associated eigenvalues $\lambda_1, \dots, \lambda_r$ are pairwise different, we suppose there is given a linear relation

$$a_1\mathbf{v}_1 + \cdots + a_r\mathbf{v}_r = \mathbf{0} \tag{24.7.1}$$

in $\mathbf{R}^n$ for scalars $a_1, \dots, a_r$ and aim to show that every a_i vanishes. In the case $r = 1$ there is nothing to do: we have $a_1\mathbf{v}_1 = \mathbf{0}$ with $\mathbf{v}_1$ nonzero (since eigenvectors are nonzero by definition), so necessarily $a_1 = 0$ as desired. Hence, we may focus now on cases with $r > 1$.

We shall proceed by the method of mathematical induction on r: we *suppose* the result is already known to hold for any collection of *less* than r eigenvectors with pairwise different eigenvalues, and aim to deduce from this fact alone that it also holds for any collection of r such vectors. Once such a general deduction is carried out, we could conclude that *if* the desired result is then really established in general (and not just assumed) for any collection of $r-1$ such vectors then it holds in general for any collection of r such vectors. In other words, we could bootstrap from settling the case for all collections of some number of such vectors to the case of all collections with one more such vector. Beginning with the known "base case" of a single eigenvector, we could then bootstrap to conclude the case $r = 2$, and then conclude the case $r = 3$, and so on to conclude the general case.

---

[23] George Green (1793-1841) was a British mathematical physicist who introduced potential functions into physics and defined "Green's functions" that became fundamental for quantum mechanics and differential equations. Forced to work full-time during ages 5 to 36 (while his father lived), he was entirely self-educated. He kept up with European math research, self-published his discoveries, and began college at Cambridge at age 39. His work was appreciated only posthumously.

So our task is to show for $r > 1$ that *if* the result is known for every collection of $r-1$ eigenvectors with pairwise different eigenvalues then it holds for our given arbitrary collection $\mathbf{v}_1, \dots, \mathbf{v}_r$ of r eigenvectors with pairwise different eigenvalues. For any scalar c, we compute the matrix-vector product $(A - c\mathrm{I}_n)\mathbf{v}_j = A\mathbf{v}_j - c\mathrm{I}_n\mathbf{v}_j = \lambda_j\mathbf{v}_j - c\mathbf{v}_j = (\lambda_j - c)\mathbf{v}_j$. By linearity in $\mathbf{v}$ for the matrix-vector product $B\mathbf{v}$ for any $n \times n$ matrix B, multiplying both sides of (24.7.1) by $A - c\mathrm{I}_n$ yields

$$(A - c\mathrm{I}_n)(a_1\mathbf{v}_1) + \cdots + (A - c\mathrm{I}_n)(a_r\mathbf{v}_r) = (A - c\mathrm{I}_n)(a_1\mathbf{v}_1 + \cdots + a_r\mathbf{v}_r) = (A - c\mathrm{I}_n)(\mathbf{0}) = \mathbf{0}.$$

Since $(A - c\mathrm{I}_n)(a_j\mathbf{v}_j) = a_j((A - c\mathrm{I}_n)\mathbf{v}_j) = a_j((\lambda_j - c)\mathbf{v}_j) = (a_j(\lambda_j - c))\mathbf{v}_j$, we deduce that

$$(a_1(\lambda_1 - c))\mathbf{v}_1 + \cdots + (a_r(\lambda_r - c))\mathbf{v}_r = \mathbf{0}.$$

This holds for all scalars c, so by choosing $c = \lambda_r$ we eliminate the final term on the left side (since the coefficient $\lambda_r - c$ becomes 0), getting $a_1(\lambda_1 - \lambda_r)\mathbf{v}_1 + \cdots + a_{r-1}(\lambda_{r-1} - \lambda_r)\mathbf{v}_{r-1} = \mathbf{0}$. This is a linear relation for the $r-1$ eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_{r-1}$. Recall we are assuming the desired result is already known for collections of $r-1$ eigenvectors with pairwise different eigenvalues, so the scalars $a_j(\lambda_j - \lambda_r)$ for $1 \leq j < r$ vanish. But $\lambda_j - \lambda_r \neq 0$ for $j < r$ since the eigenvalues are pairwise different, so we can divide by that to get $a_j = 0$ for all $j < r$. Going back to the initial linear relation $a_1\mathbf{v}_1 + \cdots + a_r\mathbf{v}_r = \mathbf{0}$, the first $r-1$ coefficients have been shown to vanish. Hence, this relation collapses to say $a_r\mathbf{v}_r = \mathbf{0}$, so since $\mathbf{v}_r$ is nonzero (as eigenvectors are nonzero by definition) we obtain that $a_r = 0$. We have shown that all coefficients a_i vanish, completing the proof of (i).

Now we turn to the proof of (ii), which will be a clever computation. Assuming A is a *symmetric* $n \times n$ matrix and $\mathbf{v}, \mathbf{v}'$ are n-vectors satisfying $A\mathbf{v} = \lambda\mathbf{v}$ and $A\mathbf{v}' = \lambda'\mathbf{v}'$ with $\lambda \neq \lambda'$, we want to show $\mathbf{v} \cdot \mathbf{v}' = 0$. By the nice interaction of symmetric matrices with dot products (Theorem 20.1.4), $\lambda(\mathbf{v} \cdot \mathbf{v}') = (\lambda\mathbf{v}) \cdot \mathbf{v}' = (A\mathbf{v}) \cdot \mathbf{v}' = \mathbf{v} \cdot (A\mathbf{v}') = \mathbf{v} \cdot (\lambda'\mathbf{v}') = \lambda'(\mathbf{v} \cdot \mathbf{v}')$. Comparing outer terms, we thereby obtain $0 = \lambda(\mathbf{v} \cdot \mathbf{v}') - \lambda'(\mathbf{v} \cdot \mathbf{v}') = (\lambda - \lambda')(\mathbf{v} \cdot \mathbf{v}')$. On the right side, the scalar $\lambda - \lambda'$ is nonzero because we assumed $\lambda \neq \lambda'$. Hence, the other factor $\mathbf{v} \cdot \mathbf{v}'$ must vanish, as desired. ☐

Having completed the promised proof, next we explain where (24.4.4) comes from. For ease of notation, let $\mathbf{w}'_j = \mathbf{w}_j / \|\mathbf{w}_j\|$; this is a unit vector along the same line as $\mathbf{w}_j$, and is also an eigenvector of A with the same eigenvalue λ_j. Hence, we have the eigenvector equations

$$A\mathbf{w}'_1 = \lambda_1\mathbf{w}'_1,\ A\mathbf{w}'_2 = \lambda_2\mathbf{w}'_2,\ \dots,\ A\mathbf{w}'_n = \lambda_n\mathbf{w}'_n.$$

We can put together all of these vector equations into a matrix equation:

$$AW = \begin{bmatrix} | & | & & | \\ A\mathbf{w}'_1 & A\mathbf{w}'_2 & \cdots & A\mathbf{w}'_n \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \lambda_1\mathbf{w}'_1 & \lambda_2\mathbf{w}'_2 & \cdots & \lambda_n\mathbf{w}'_n \\ | & | & & | \end{bmatrix} = W \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}.$$

Now multiply both sides on the right by $W^{-1} = W^\top$ to get the desired result (since $(AW)W^{-1} = A(WW^{-1}) = A\mathrm{I}_n = A$).

Chapter 24 highlights (links to highlights in and next chapters)

Notation	**Meaning**	**Location in text**
nothing new!		

Concept	**Meaning**	**Location in text**
pos-definite, neg-definite, indefinite, pos/neg-semidefinite for $n \times n$ symmetric matrix A	these notions encode sign(s) of values of $q_A(\mathbf{v})$ for varying nonzero $\mathbf{v} \in \mathbf{R}^n$	Definition 24.2.2
dominant eigenvalue for $n \times n$ symmetric matrix A	eigenvalue λ for which $\lvert\lambda\rvert$ is larger than absolute value of all other eigenvalues	Proposition 24.4.2

Result	**Meaning**	**Location in text**
for $n \times n$ matrix A, eigenvectors with distinct eigenvalues are linearly independent	if $\mathbf{v}_1, \dots, \mathbf{v}_k$ satisfy $A\mathbf{v}_i = \lambda_i \mathbf{v}_i$ with pairwise distinct λ_i's then the $\mathbf{v}_i$'s are linearly independent	Theorem 24.1.1(i)
for $n \times n$ **symmetric** A, eigenspaces for distinct eigenvalues are orthogonal to each other	if $A\mathbf{v} = \lambda\mathbf{v}$ and $A\mathbf{w} = \mu\mathbf{w}$ with $\lambda \neq \mu$ then $\mathbf{v} \cdot \mathbf{w} = 0$	Theorem 24.1.1(ii)
Spectral Theorem	for any $n \times n$ symmetric matrix A, there is an orthogonal basis of $\mathbf{R}^n$ consisting of eigenvectors for A	Theorem 24.1.4
diagonalization formula for $q_A(\mathbf{x})$ for $n \times n$ symmetric A with orthogonal basis $\mathbf{w}_1, \dots$ of eigenvectors satisfying $A\mathbf{w}_i = \lambda_i \mathbf{w}_i$	when $\mathbf{x} \in \mathbf{R}^n$ is *written in terms of* $\mathbf{w}_i$*'s* as $\sum t_i \mathbf{w}_i$ then $q_A(\mathbf{x})$ has no $t_i t_j$ cross-terms $(i \neq j)$ and has t_i^2-coefficient $\lambda_i(\mathbf{w}_i \cdot \mathbf{w}_i)$	(24.2.2)
eigenvectors interact well with powers and inversion	if $A\mathbf{v} = \lambda\mathbf{v}$ then $A^r\mathbf{v} = \lambda^r\mathbf{v}$ for all $r \geq 1$, and also for $r = -1$ if A invertible	(24.3.1), Remark 24.3.1
efficiently compute powers of an $n \times n$ symmetric matrix in terms of eigenvalues and eigenvectors	if $\mathbf{w}_1, \dots,$ is orthogonal basis of eigenvectors and $\lambda_1, \dots$ are the eigenvalues then $A^m = WD^mW^\top$ for all $m \geq 1$, where D is diagonal with $d_{ii} = \lambda_i$ and W is orthogonal with jth column $\mathbf{w}_j / \lVert\mathbf{w}_j\rVert$	Theorem 24.4.1, (24.4.5)
for $n \times n$ symmetric A, dominant eigenvalue (if one exists) controls A^m for big m	if A has a dominant eigenvalue λ and the λ-eigenspace is a **line**, spanned by $\mathbf{w}$, then $A^m \approx (\lambda^m/(\mathbf{w} \cdot \mathbf{w}))\mathbf{w}\mathbf{w}^\top$ for big m	Proposition 24.4.2 (do not need to know for exams!)

Skill	**Location in text**
given an orthogonal basis $\mathbf{w}_1, \dots, \mathbf{w}_n$ of eigenvectors for an $n \times n$ symmetric A, express $q_A(\mathbf{x})$ with no cross-terms upon *writing* $\mathbf{x}$ *in terms of the* $\mathbf{w}_i$*'s*	Examples 24.2.8, 24.2.9
read off definiteness of an $n \times n$ symmetric matrix by inspecting signs of eigenvalues (**not** by inspecting signs of matrix entries!)	Prop. 24.2.10, Rem. 24.2.11
for 2×2 symmetric A, compute $A^r(\mathbf{v})$ for $r \geq 1$ (and $r = -1$ if A invertible) by *writing* $\mathbf{v}$ *in terms of an orthogonal basis of eigenvectors* for A	Example 24.3.2

25. The Hessian and quadratic approximation

In single-variable calculus we use the second derivative f'' of a function $f : \mathbf{R} \to \mathbf{R}$ to analyze when a critical point (i.e., a point where f' vanishes) is a local maximum or local minimum. For any a at which f' vanishes, if $f''(a) < 0$ then $x = a$ is a local maximum for f (i.e., $f(a) \geq f(x)$ for x near a) and if $f''(a) > 0$ then $x = a$ is a local minimum for f (i.e., $f(a) \leq f(x)$ for x near a); this is the "second derivative test" from single-variable calculus (which is inconclusive when $f''(a) = 0$).

We now work towards an analogue of this concept for a scalar-valued multivariable function $f : \mathbf{R}^n \to \mathbf{R}$. The main wrinkle, as we saw in Chapter 9 and Example 10.2.11, is that there are *several* second derivatives at each $\mathbf{a} \in \mathbf{R}^n$ (ultimately due to the greater range of possibilities for how to move around in $\mathbf{R}^n$ for $n > 1$, in contrast with motion in $\mathbf{R}$). For example, a function $f(x, y)$ of 2 variables has three associated "second derivatives" at each $(a, b) \in \mathbf{R}^2$.

We will arrange these second derivatives as the entries in a (symmetric) matrix $(\mathrm{H}f)(\mathbf{a})$, called the *Hessian*, and use it to make accurate approximations to f near $\mathbf{a}$. In Chapter 26 we will use the Hessian matrix of second derivatives to *systematically* find local maxima and local minima for multivariable functions $f : \mathbf{R}^n \to \mathbf{R}$ (recovering the second-derivative test from single-variable calculus when $n = 1$).

By the end of this chapter, you should be able to:

- symbolically compute the Hessian $(\mathrm{H}f)(\mathbf{a})$ at $\mathbf{a} \in \mathbf{R}^n$ for a multivariable function $f : \mathbf{R}^n \to \mathbf{R}$;
- determine the quadratic approximation to f near $\mathbf{a} \in \mathbf{R}^n$ in terms of $(\nabla f)(\mathbf{a})$ and $(\mathrm{H}f)(\mathbf{a})$;
- draw approximate level curves of $q_A(\mathbf{x})$ for symmetric 2×2 matrices A with nonzero eigenvalues, and relate this to level curves of $f(x, y)$ near a critical point $\mathbf{a} \in \mathbf{R}^2$ with $A = (\mathrm{H}f)(\mathbf{a})$.

25.1. Review of second partials. In Section 9.6 we introduced second partial derivatives. As a review of computing second partials and the surprising fact called "equality of mixed partials" (Theorem 9.6.4), let's work out a 3-variable example to see this all in action.

Example 25.1.1. Consider $f(x, y, z) = 5x^2y - 7z^3 + xe^{2y}\sin(z) + 2\,\frac{y}{z} - \ln(xy + z)$. We have

$$f_x = 10xy + e^{2y}\sin(z) - \frac{y}{xy+z}, \quad f_y = 5x^2 + 2xe^{2y}\sin(z) + \frac{2}{z} - \frac{x}{xy+z},$$

$$f_z = -21z^2 + xe^{2y}\cos(z) - 2\,\frac{y}{z^2} - \frac{1}{xy+z}.$$

Thus (please check for yourself),

$$\frac{\partial^2 f}{\partial y \partial x} = 10x + 2e^{2y}\sin(z) - \frac{(xy+z) - yx}{(xy+z)^2}, \quad \frac{\partial^2 f}{\partial x \partial y} = 10x + 2e^{2y}\sin(z) - \frac{(xy+z) - xy}{(xy+z)^2}, \tag{25.1.1}$$

so $\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x}$ by inspection. The intermediate steps for computing $\frac{\partial^2 f}{\partial x \partial y}$ and $\frac{\partial^2 f}{\partial y \partial x}$ are *not* the same (one of them passes through computing f_x, the other through computing f_y, and the formulas for f_x and f_y are rather different from each other), so it may be surprising that the final results coincide (as they must, by Theorem 9.6.4).

Likewise, $\frac{\partial^2 f}{\partial z \partial x} = e^{2y}\cos(z) + \frac{y}{(xy+z)^2}$ and $\frac{\partial^2 f}{\partial x \partial z} = e^{2y}\cos(z) + \frac{y}{(xy+z)^2}$ (please check those for yourself), so $\frac{\partial^2 f}{\partial z \partial x} = \frac{\partial^2 f}{\partial x \partial z}$ by inspection. Finally (please check for yourself),

$$\frac{\partial^2 f}{\partial z \partial y} = 2xe^{2y}\cos(z) - \frac{2}{z^2} + \frac{x}{(xy+z)^2}, \quad \frac{\partial^2 f}{\partial y \partial z} = 2xe^{2y}\cos(z) - \frac{2}{z^2} + \frac{x}{(xy+z)^2},$$

so by inspection the equality of mixed partials $\dfrac{\partial^2 f}{\partial z \partial y} = \dfrac{\partial^2 f}{\partial y \partial z}$ also holds. ■

One can also define and compute higher-order partial derivatives, such as

$$\frac{\partial^3 g}{\partial y \partial x \partial w}, \quad \frac{\partial^4 g}{\partial x \partial y \partial y \partial w}$$

for a function $g(x, y, z, w)$. **Equality of mixed partials holds in the higher-order case too** (rearranging the partial derivatives in any order); it can be deduced from the case of equality of second-order partials in Theorem 9.6.4. We will not investigate those in this course, but they arise in an essential way in multivariable power series (generalizing single-variable power series) and in work on *partial differential equations*: these involve relations among partial derivatives of functions of more than one variable.

With equality of second-order mixed partials (and even for higher-order mixed partials) in our mathematical toolkit, we introduce some shorthand: for $f(x_1, \dots, x_n)$ and $1 \le i, j \le n$ we define the notation

$$f_{x_i x_j} = \frac{\partial^2 f}{\partial x_i \partial x_j} = f_{x_j x_i} \tag{25.1.2}$$

and similarly $g_{yxw} = \dfrac{\partial^3 g}{\partial y \partial x \partial w}$, $g_{xyyw} = \dfrac{\partial^4 g}{\partial x \partial y \partial y \partial w} = \dfrac{\partial^4 g}{\partial x \partial y^2 \partial w}$ for any $g(x, y, z, w)$, and so on.

Example 25.1.2. Much as differential equations in a single variable (typically "time") pervade many applications in the natural sciences for which one studies how a system evolves, for phenomena involving the interaction among several parameters one encounters partial differential equations.

Many equations in physics, economics, and engineering are partial differential equations; such equations also arise in computer science (especially related to images). Here are some famous examples:

- In the study of how heat dissipates through a region of space, if $u(x, y, z, t)$ is the temperature at time t at a point (x, y, z) then considerations with conservation of energy and Fourier's law of heat conduction lead to the *heat equation*

  $$u_t = \alpha(u_{xx} + u_{yy} + u_{zz})$$

  for a physical constant $\alpha > 0$ (called the "thermal diffusivity").
- For a class of financial instruments called "options", which involve a bet on the future value of an investment and have existed in various forms going back centuries, it was a long-standing problem to figure out the right way to set their price. The solution, inspired by ideas from statistical physics and leading to the 1997 Nobel Prize in economics, was given in terms of a partial differential equation discovered in the early 1970's by Fischer Black (then at the University of Chicago) and Myron Scholes (then at MIT; he later moved to the University of Chicago, and then Stanford): if $V(S, t)$ is the "best" value for an option on a stock with price S at time t then the *Black–Scholes equation* is

  $$V_t + (1/2)\sigma^2 S^2 V_{SS} = rV - rSV_S$$

  where σ is a measure of volatility of the stock and r is a certain "risk-free interest rate". (An explicit solution, called the *Black–Scholes formula*, was given by Black and Scholes; around the same time a more robust solution was discovered by Robert C. Merton, then at MIT.) Curiously, after a non-obvious change of variables this becomes the heat equation for 1 spatial dimension.
- The electric and magnetic forces arising from a distribution of electrical charge in space (which may moreover be moving in time) is governed by a system of linked partial differential equations called *Maxwell's equations*. Even when expressed in terms of the efficient language of vector calculus, the equations are quite a mouthful to write out (as one sees on various t-shirts).

- In biology, the *Fisher-KPP equation*

$$u_t = D(u_{xx} + u_{yy}) + ru(1 - u/K)$$

governs the population density $u(x, y, t)$ of an advantageous gene at time t and position (x, y) (for constants D, r, and K related to the biological situation).

There are many more examples in physics (e.g., Schrödinger's equation in quantum mechanics, Einstein's field equations in General Relativity), image processing and computer vision (the Perona–Malik equation to smooth or restore images), fluid mechanics (the Navier–Stokes equation, also relevant to weather forecasting, aerodynamics, and realistic-looking smoke in video games), and so on.

■

Remark 25.1.3. There is a surprising dichotomy between single-variable differential equations (called "ordinary" differential equations and studied in Math 53) and partial differential equations. Although not relevant to this course, it is worthwhile to be aware of the distinction: for ordinary differential equations essentially all examples have solutions (with any initial condition) but for partial differential equations this is not true. In fact, there is essentially only one general result guaranteeing existence of a solution for a *large class* of examples: the equations expressing a time derivative $(\partial/\partial t)^k f$ in terms of a mixture of spatial and time derivatives with total order at most k. This result, the (higher-order) Cauchy–Kovalevskaya[24] Theorem, has a restrictive hypothesis (called "analyticity") that limits its applicability, so most results on partial differential equations focus on *specific* equations rather than on large classes of equations.

For example, existence and uniqueness for solutions to Einstein's equations in General Relativity was first proved in useful generality in 1951 by Choquet-Bruhat[25], around 35 years *after* Einstein's discovery of the equations in 1915. Also, the existence and uniqueness (under physically reasonable initial conditions) for solutions to the 2-dimensional case of the Navier-Stokes equations in fluid mechanics was first proved in the late 1950's largely by Ladyzhenskaya[26]; the 3-dimensional case remains wide open and is one of the million-dollar Clay Millenium Problems.

25.2. The Hessian matrix. The role of the second derivative at a point, for multivariable functions $f : \mathbf{R}^n \to \mathbf{R}$, is played by an $n \times n$ (square) matrix called the *Hessian*. [27] For $n = 1$ this is the 1×1 matrix whose single entry is the scalar value of the second derivative from single-variable calculus.

Let's discuss the case $n = 2$ before we move onto the general case. For a function $f(x, y)$ of two variables, there are four types of second partial derivative, two of which are equal:

$$\frac{\partial^2 f}{\partial x^2}, \quad \frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x}, \quad \frac{\partial^2 f}{\partial y^2}.$$

---

[24] Sofia Kovalevskaya (1850-1891) was a Russian mathematician who worked on differential equations and mathematical problems in mechanics. Her PhD thesis included the first general existence theorem for solvability of partial differential equations (known earlier only in special cases), and during her academic career in Stockholm she produced prize-winning work on rigid body mechanics. She began to learn about calculus at age 11 when her parents compensated for a shortage of wallpaper by using her father's old notes from his college calculus course as wallpaper for her room (a novel type of "applied math").

[25] Yvonne Choquet-Bruhat (b. 1923) is a French mathematician who has had a prolific career of more than a half-century at the forefront of mathematical physics. Her scientific work, which led to membership in the French Academy of Sciences in 1979, began in the early 1950's when she established breakthrough existence theorems in mathematical General Relativity. She worked in Princeton during 1951-52, where she discussed her ideas with Einstein, an experience that she summarizes in [C].

[26] Olga Ladyzhenskaya (1922-2004) emerged from tragic family circumstances in Stalinist Russia (her father was executed as an "enemy of the people" when she was 15, due to which she was initially banned from university education) to become one of the world's leading experts in partial differential equations in the mid-20th century. She was on the shortlist for the Fields Medal in 1958, and led a large research group in mathematical physics at the Steklov Institute in St. Petersburg, where she was a Director for nearly 40 years.

[27] Otto Hesse (1811-1874) was a German mathematician who worked on algebraic geometry and the calculus of variations.

We put these into a 2×2 matrix $\begin{bmatrix} \dfrac{\partial^2 f}{\partial x^2} & \dfrac{\partial^2 f}{\partial x \partial y} \\ \dfrac{\partial^2 f}{\partial y \partial x} & \dfrac{\partial^2 f}{\partial y^2} \end{bmatrix}$ that we call the *Hessian*; this is symmetric precisely because $\dfrac{\partial^2 f}{\partial x \partial y} = \dfrac{\partial^2 f}{\partial y \partial x}$. Each of the entries here is a function that we can evaluate at a common point in $\mathbf{R}^2$ to get a 2×2 matrix with scalar entries.

For example, if $f(x, y) = x^3 + xy^2 - 5y^4$ then $f_x = 3x^2 + y^2$ and $f_y = 2xy - 20y^3$, so $f_{xx} = 6x$, $f_{xy} = f_{yx} = 2y$, and $f_{yy} = 2x - 60y^2$. The Hessian at a point $(x, y) = (a, b)$ is then

$$\begin{bmatrix} 6a & 2b \\ 2b & 2a - 60b^2 \end{bmatrix}.$$

Note that this is indeed symmetric. Now for the general case:

Definition 25.2.1. For a function $f : \mathbf{R}^n \to \mathbf{R}$, the *Hessian matrix* $(\mathrm{H}f)(\mathbf{a})$ of f at a point $\mathbf{a} \in \mathbf{R}^n$ is defined to be the matrix of second partial derivatives, with ij-entry equal to $\dfrac{\partial^2 f}{\partial x_i \partial x_j}(\mathbf{a})$ (so there is nothing complicated to memorize: the entry label matches the second partial occurring there):

$$(\mathrm{H}f)(\mathbf{a}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(\mathbf{a}) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(\mathbf{a}) & \frac{\partial^2 f}{\partial x_1 \partial x_3}(\mathbf{a}) & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(\mathbf{a}) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(\mathbf{a}) & \frac{\partial^2 f}{\partial x_2^2}(\mathbf{a}) & \frac{\partial^2 f}{\partial x_2 \partial x_3}(\mathbf{a}) & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(\mathbf{a}) \\ \frac{\partial^2 f}{\partial x_3 \partial x_1}(\mathbf{a}) & \frac{\partial^2 f}{\partial x_3 \partial x_2}(\mathbf{a}) & \frac{\partial^2 f}{\partial x_3^2}(\mathbf{a}) & \dots & \frac{\partial^2 f}{\partial x_3 \partial x_n}(\mathbf{a}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(\mathbf{a}) & \frac{\partial^2 f}{\partial x_n \partial x_2}(\mathbf{a}) & \frac{\partial^2 f}{\partial x_n \partial x_3}(\mathbf{a}) & \dots & \frac{\partial^2 f}{\partial x_n^2}(\mathbf{a}) \end{bmatrix}.$$

Remark 25.2.2. The Hessian matrix $(\mathrm{H}f)(\mathbf{a})$ of second partial derivatives is exactly $(D(\nabla f))(\mathbf{a})$ from Example 13.5.6, which is *symmetric* due to equality of mixed partials. In Section 25.3 the Hessian $(\mathrm{H}f)(\mathbf{a})$ will be used to analyze the behavior of $f(x_1, \dots, x_n)$ near a point $\mathbf{a}$, to be built upon in Chapter 26 via the Spectral Theorem for the (symmetric!) Hessian when $\mathbf{a}$ is a critical point, and in Section 25.4 its visual significance will be illustrated for the case $n = 2$.

Example 25.2.3. For $F(x, y, z) = x^2yz + \cos(xy) + ze^y$, the Hessian $(\mathrm{H}F)(x, y, z)$ is in (13.5.2). ■

Example 25.2.4. Let's compute the gradient and the Hessian of the function $f(x, y) = \ln(xy - 1)$ at the point $(x, y) = (1, 2)$. The partial derivatives are $f_x = \dfrac{y}{xy - 1}$ and $f_y = \dfrac{x}{xy - 1}$, so $(\nabla f)(1, 2) = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$. From the first partial derivatives we compute the second partial derivatives:

$$f_{xx} = \frac{-y^2}{(xy - 1)^2}, \quad f_{yy} = \frac{-x^2}{(xy - 1)^2}, \quad f_{xy} = \frac{-1}{(xy - 1)^2}.$$

Sticking these into a matrix yields

$$(\mathrm{H}f)(x, y) = \begin{bmatrix} -y^2/(xy - 1)^2 & -1/(xy - 1)^2 \\ -1/(xy - 1)^2 & -x^2/(xy - 1)^2 \end{bmatrix},$$

and then plugging in the point $(1, 2)$ gives $(\mathrm{H}f)(1, 2) = \begin{bmatrix} -4 & -1 \\ -1 & -1 \end{bmatrix}$. ■

Remark 25.2.5. The Hessian arises in many real-world optimization problems. We discuss this in the context of Newton's method for optimization in Appendix I (useful for speeding up gradient descent algorithms that arise in machine learning and solving non-linear problems in many scientific and engineering contexts) and for applications in economics and the natural sciences in Chapter 26 (see Examples 26.2.4, 26.2.5, and 26.4.3) and Appendix J. Section 25.3 discusses its role in improving linear approximation via gradients, leading to the multivariable second derivative test in Chapter 26.

25.3. Using the Hessian for approximations.

Definition 25.3.1. The *quadratic approximation* to $f : \mathbf{R}^n \to \mathbf{R}$ near $\mathbf{a} \in \mathbf{R}^n$ is the expression on the right side of

$$f(\mathbf{a}+\mathbf{h}) \approx f(\mathbf{a}) + (\nabla f)(\mathbf{a}) \cdot \mathbf{h} + \frac{1}{2}\mathbf{h}^\top((\mathrm{H}f)(\mathbf{a}))\mathbf{h} \tag{25.3.1}$$

(with small $\mathbf{h}$), where the last term uses matrix-vector products; this refines "linear approximation" via the gradient from (11.1.2).

(The single-variable analogue is $f(a+h) \approx f(a) + f'(a)h + \frac{1}{2}f''(a)h^2$, so $(\mathrm{H}f)(\mathbf{a})$ plays the role that $f''(a)$ does; for $n = 1$ the matrix $(\mathrm{H}f)(\mathbf{a})$ is the 1×1 matrix $[f''(a)]$.)

In Section 25.5 we explain where this approximation comes from, for those who are interested. It is the "degree 2" part of a multivariable Taylor series for f at $\mathbf{a}$ (there is a general theory of multivariable power series, which we are not discussing in this course), so the factor $1/2$ appears for the same reason that $1/2! = 1/2$ appears in the quadratic part of a Taylor series in single-variable calculus.

Example 25.3.2. Continuing Example 25.2.4, let's use the gradient and Hessian to estimate the value of $f(x,y) = \ln(xy-1)$ at $(1.2, 1.8) = \mathbf{a} + \mathbf{h}$ for $\mathbf{a} = (1,2)$ and $\mathbf{h} = (0.2, -0.2)$.

We already computed that $(\nabla f)(1,2) = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ and $(\mathrm{H}f)(1,2) = \begin{bmatrix} -4 & -1 \\ -1 & -1 \end{bmatrix}$. Also $f(1,2) = 0$, so for all small $(h,k) \in \mathbf{R}^2$ the quadratic approximation to $f(1+h, 2+k)$ is

$$f(1+h, 2+k) \approx 0 + \begin{bmatrix} 2 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} h \\ k \end{bmatrix} + \frac{1}{2}\begin{bmatrix} h \\ k \end{bmatrix}^\top \begin{bmatrix} -4 & -1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} h \\ k \end{bmatrix} = 2h + k + \frac{1}{2}\left(-4h^2 - 2hk - k^2\right).$$

To estimate $f(1.2, 1.8)$ we take $(h,k) = (0.2, -0.2)$ to get $f(1.2, 1.8) \approx 0.14$. In reality, $f(1.2, 1.8) = 0.1484\ldots$; the agreement is much better than the linear approximation, which gives $f(1.2, 1.8) \approx 0.2$. ■

Example 25.3.3. Let $f(x,y) = \cos(x/y)$. We use the gradient and Hessian to estimate the value of $f(x,y)$ near $\mathbf{a} = (0,1)$. By computing partial derivatives of f we obtain that

$$(\nabla f)(x,y) = \begin{bmatrix} -(1/y)\sin(x/y) \\ (x/y^2)\sin(x/y) \end{bmatrix}.$$

Computing partial derivatives of each entry in the gradient, we obtain second partial derivatives and hence:

$$(\mathrm{H}f)(x,y) = \begin{bmatrix} -\cos(x/y)/y^2 & (1/y^3)(y\sin(x/y) + x\cos(x/y)) \\ (1/y^3)(y\sin(x/y) + x\cos(x/y)) & -(x/y^4)(2y\sin(x/y) + x\cos(x/y)) \end{bmatrix}.$$

Thus, the quadratic approximation

$$f(\mathbf{a}+\mathbf{h}) \approx f(\mathbf{a}) + (\nabla f)(\mathbf{a}) \cdot \mathbf{h} + \frac{1}{2}\mathbf{h}^\top((\mathrm{H}f)(\mathbf{a}))\mathbf{h}$$

at $\mathbf{a} = (0,1)$ says that for (h_1, h_2) near $(0,0)$,

$$\begin{aligned}\cos(h_1/(1+h_2)) = f(h_1, 1+h_2) &\approx f(0,1) + (\nabla f)(0,1)\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} h_1 & h_2 \end{bmatrix}((\mathrm{H}f)(0,1))\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \\ &= 1 + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} h_1 & h_2 \end{bmatrix}\begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \\ &= 1 + 0 - \frac{1}{2}h_1^2 \\ &= 1 - \frac{1}{2}h_1^2\end{aligned}$$

■

Example 25.3.4. The scale on which the quadratic approximation applies can be *very small*, even for an innocuous-looking function. Consider $f(x,y) = 2x^2 - 5xy + 2y^3 + 3x^2y + 2y^2$ whose surface graph across the square region of points (x,y) with $-2 \le x, y \le 2$ is shown in Figure 25.3.1 (the height of the graph is on a rather condensed scale, since $f(-2,2) = 76$).

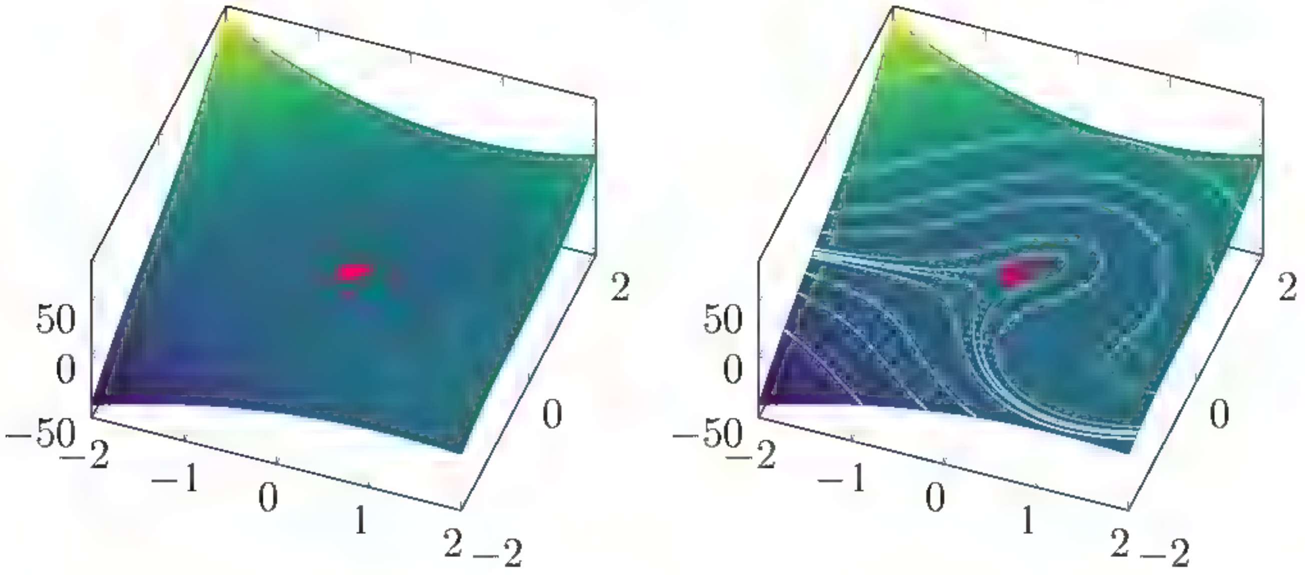


FIGURE 25.3.1. Surface graph of $f(x,y) = 2x^2 - 5xy + 2y^3 + 3x^2y + 2y^2$ with two rather close critical points, one a local minimum and one a saddle point; which is which? The white curves in the picture on the right are level curves for f.

There are two critical points for f, corresponding to two red dots in the surface graph: one red dot lies over $P = (0,0)$, and the other lies over $Q = (a,b) \approx (0.137369, 0.131586)$. (To be precise, a is the unique real root of $36x^3 - 120x^2 + 125x - 15$ and $b = 4a/(5-6a)$: the latter expresses that $f_x(a,b) = 0$ since $f_x = 4x - (5-6x)y$ with $a \neq 0$, and plugging this into the equation $f_y(a,b) = 0$ gives the cubic condition on $a \neq 0$.) It is hard to tell from the scale of the picture that the local behavior of f near these points is *completely different*: P is a saddle point and Q is a local minimum.

In Chapter 26 we will learn systematic techniques based on eigenvalues of Hessians to figure out the behavior near a critical point, and we will revisit this example and zoom in by a factor of 25 to see what is going on near these points. In particular, we'll see the very different nature of the contour plots sufficiently near each of these points. (If you want to kill the suspense, see Example 26.4.5 now.) The main lesson is that the usefulness of the quadratic approximation discussed above may require working on an extremely small region near a critical point, as in this example (where the types of the quadratic approximation at the saddle point P and local minimum Q are very different). ■

25.4. Geometry of contour plots near critical points. The quadratic approximation (25.3.1) can be used to understand the behavior of a function $f : \mathbf{R}^2 \to \mathbf{R}$ near a critical point (e.g., local maximum, local minimum, or saddle point?) in terms of the contour plot near such a point. The link between the geometry of the contour plot near such a point and the behavior of f near there was suggested in Example 10.2.13, where it seemed that the level sets of f near a critical point are approximate ovals near a local extremum and are approximate hyperbolas approaching an "X" near a saddle point. The quadratic approximation makes it possible to understand that observation in more precise terms, as we now explain.

The basic idea is as follows. Suppose $\mathbf{a} \in \mathbf{R}^2$ is a *critical point* of $f : \mathbf{R}^2 \to \mathbf{R}$, so $(\nabla f)(\mathbf{a}) = \mathbf{0}$. Hence, the gradient term in the quadratic approximation (25.3.1) disappears, so for $\mathbf{h}$ near $\mathbf{0}$ we have

$$f(\mathbf{a}+\mathbf{h}) \approx f(\mathbf{a}) + \frac{1}{2} q_{(\mathrm{H}f)(\mathbf{a})}(\mathbf{h}),$$

where $q_A(\mathbf{x})$ is the quadratic form $\mathbf{x}^\top A\mathbf{x}$ for a symmetric $n \times n$ matrix A and $\mathbf{x} \in \mathbf{R}^n$. Hence, for a scalar c near $f(\mathbf{a})$, the level curve $f(\mathbf{a}+\mathbf{h}) = c$ for $\mathbf{h}$ near $\mathbf{0}$ is well-approximated by the level curve $f(\mathbf{a}) + (1/2)q_{(\mathrm{H}f)(\mathbf{a})}(\mathbf{h}) = c$ for $\mathbf{h}$ near $\mathbf{0}$, or equivalently is by $q_{(\mathrm{H}f)(\mathbf{a})}(\mathbf{h}) = 2(c - f(\mathbf{a}))$ for $\mathbf{h}$ near $\mathbf{0}$.

In other words, up to replacing the scalar $c \approx f(\mathbf{a})$ with the scalar $2(c - f(\mathbf{a})) \approx 0$ and shifting the coordinate system to be centered at $(0,0)$ rather than at $\mathbf{a}$, level curves for f near $\mathbf{a}$ are approximated by level curves for $q_{(\mathrm{H}f)(\mathbf{a})}(\mathbf{x})$ near $(0,0)$. Thus, *the level curves of f near a critical point $\mathbf{a} \in \mathbf{R}^2$ are well-approximated by the level curves of the quadratic form $q_{(\mathrm{H}f)(\mathbf{a})}$ near the origin.* This motivates:

Goal. For a symmetric 2×2 matrix A, describe the level curves of $q_A(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x}$ for $\mathbf{x}$ near $(0,0)$.

We shall use geometric information from the Spectral Theorem, such as orthogonal eigenvectors and their associated eigenvalues, to achieve this Goal. In Chapter 26 that will be applied to $A = (\mathrm{H}f)(\mathbf{a})$ to turn such geometric information into a description of how $f : \mathbf{R}^2 \to \mathbf{R}$ behaves near a critical point $\mathbf{a}$ (and further use of the Spectral Theorem applied to Hessians will provide a way to analyze n-variable functions near critical points for any n). For the rest of this section, we focus on the preceding Goal that we have seen is well-motivated by the task of understanding how 2-variable functions behave near critical points.

As a preliminary step, we discuss the graphs of certain equations as hyperbolas, determining their asymptotes and where they meet the coordinate axes. There is no need to memorize any general formulas, since the numerical example will illustrate how to work out the geometry in each case from scratch.

Example 25.4.1. Let's figure out the graph of $4t_1^2 - 3t_2^2 = 7$. As for any equation of the form $At_1^2 - Bt_2^2 = C$ with $A, B > 0$ and $C \neq 0$, this is a hyperbola aligned with the coordinate axes.

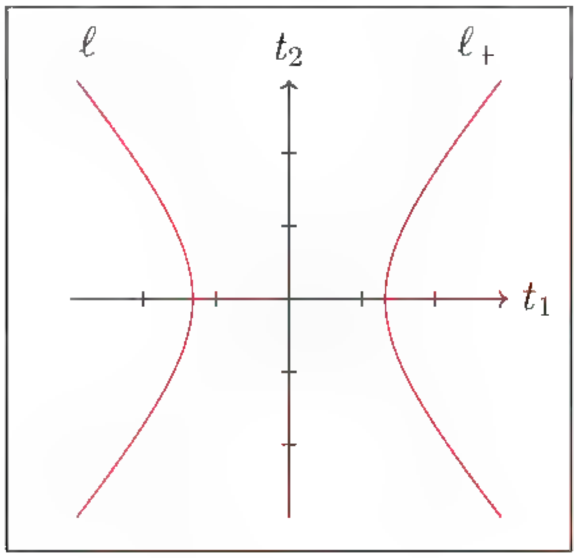


FIGURE 25.4.1. A sketch of $4t_1^2 - 3t_2^2 = 7$, with asymptotes $\ell_\pm$ as dotted lines a bit nearer to the t_2-axis than to the t_1-axis, where $\ell_\pm$ is the line $t_2 = \pm\sqrt{4/3}\,t_1$ with slope $\pm\sqrt{4/3}$.

We shall use basic algebra to get the picture in Figure 25.4.1 (though we will only care about some qualitative features that can be determined by human thought, not needing any numerical precision from a computer). To determine algebraically which of the axis lines it crosses, we set each t_j equal to 0 in the equation and try to solve for the remaining variable. If we set $t_1 = 0$ ("t_2-axis") we get $-3t_2^2 = 7$, which is impossible due to the incompatible signs of the two sides. So the graph doesn't meet that line. If we set $t_2 = 0$ ("t_1-axis") then we get $4t_1^2 = 7$, or $t_1 = \pm\sqrt{7/4}$. So the graph cuts the t_1-axis in the two points $t_1 = \pm\sqrt{7/4}$ that lie somewhere between ± 1 and ± 2 (since $7/4$ is between 1 and $4 = 2^2$). We shall next work out the precise asymptotes (i.e., the slopes of those two lines), though for qualitative purposes we will only care if the slopes have absolute value bigger or smaller than (or equal to) 1.

To find the asymptotes, which are lines that approximate the curve very well far from the origin, divide the equation "$4t_1^2 - 3t_2^2 = 7$" by either t_1^2 or t_2^2; both will lead to the same equations for the asymptotes (written either as $t_2 = \pm m t_1$ for some $m \neq 0$ or as $t_1 = \pm m' t_2$ for $m' = 1/m$). If we divide by t_2^2 then we get $4(t_1/t_2)^2 - 3 = 7/t_2^2 \approx 0$ for $|t_2|$ large, so $4(t_1/t_2)^2 \approx 3$ or equivalently $t_1/t_2 \approx \pm\sqrt{3/4}$. This gives us the asymptotes as $t_1 = \pm\sqrt{3/4}\, t_2$ or equivalently $t_2 = \pm\sqrt{4/3}\, t_1$. Hence, the slopes are $\pm\sqrt{4/3}$ if we write the t_1-axis horizontally and the t_2-axis vertically as in Figure 25.4.1.

Since $|\pm\sqrt{4/3}| > 1$ (as $4/3 > 1$), these asymptotes are nearer to the t_2-axis than to the t_1-axis; think of the lines $t_2 = \pm 2t_1$ and $t_2 = \pm(1/2)t_1$ to remember which coordinate axis a line through the origin is "nearer" depending on whether the slope has absolute value > 1 or < 1 (or $= 1$). The important qualitative features are that the hyperbola crosses the t_1-axis between ± 1 and ± 2, and the asymptotes are nearer to the t_2-axis than to the t_1-axis (since the slopes $\pm\sqrt{4/3}$ are bigger than 1 in absolute value). ■

Next, we use the method of the preceding example to figure out the geometry of the level sets for a specific indefinite 2-variable quadratic form; these level sets will be "tilted hyperbolas".

Example 25.4.2. For the symmetric matrix $A = \begin{bmatrix} 1 & 2 \\ 2 & -2 \end{bmatrix}$ we seek to plot the level sets

$$q_A(x, y) = x^2 + 4xy - 2y^2 = c$$

for nonzero scalars c. Since $\det A = -6 < 0$, this is indefinite; for such cases the level sets will always be hyperbolas (as we shall soon see quite concretely). In Example 24.2.9 we worked out an orthogonal basis of eigenvectors for A: $\mathbf{w}_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$, $\mathbf{w}_2 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$ with respective eigenvalues $\lambda_1 = 2$ and $\lambda_2 = -3$.

To maintain accurate contact with geometry, consider the *unit* eigenvectors $\mathbf{w}_i' = \mathbf{w}_i / \|\mathbf{w}_i\|$. The pair $\{\mathbf{w}_1', \mathbf{w}_2'\}$ is an *orthonormal basis*, so writing vectors in terms of these is tantamount to applying a rotation to $\mathbf{R}^2$ that moves the standard orthonormal basis $\{\mathbf{e}_1, \mathbf{e}_2\}$ into this new one. In other words, by writing $\mathbf{v} = t_1'\mathbf{w}_1' + t_2'\mathbf{w}_2'$ for $\mathbf{v} \in \mathbf{R}^2$, such (t_1', t_2')-coordinates correspond to a grid that is a rotated version of the standard one. When q_A is expressed in terms of these new coordinates, by the diagonalization formula (24.2.2) the expression will have no cross-term, so the geometry will be easier to understand than in the original expression with (x, y)-coordinates.

To be specific, when (24.2.2) is applied with these unit eigenvectors we obtain

$$q_A(t_1'\mathbf{w}_1' + t_2'\mathbf{w}_2') = \lambda_1 {t_1'}^2 + \lambda_2 {t_2'}^2 = 2{t_1'}^2 - 3{t_2'}^2$$

where $\lambda_1 = 2$ is the eigenvalue for $\mathbf{w}_1'$ and $\lambda_2 = -3$ is the eigenvalue for $\mathbf{w}_2'$. The opposite signs of the eigenvalues, which are the coefficients on the right side, tell us as in Example 25.4.1 that the level curves of the quadratic form are hyperbolas whose axes of symmetry are the t_1'-axis and t_2'-axis. These two axes are the perpendicular lines spanned by $\mathbf{w}_1'$ and by $\mathbf{w}_2'$, or equivalently by $\mathbf{w}_1$ and by $\mathbf{w}_2$.

We now work out the sketch of the level set $q_A(x, y) = c$ for two values of c with opposite signs, say $c = 5$ and $c = -4$. Eventually we will arrive at the picture shown in Figure 25.4.2.

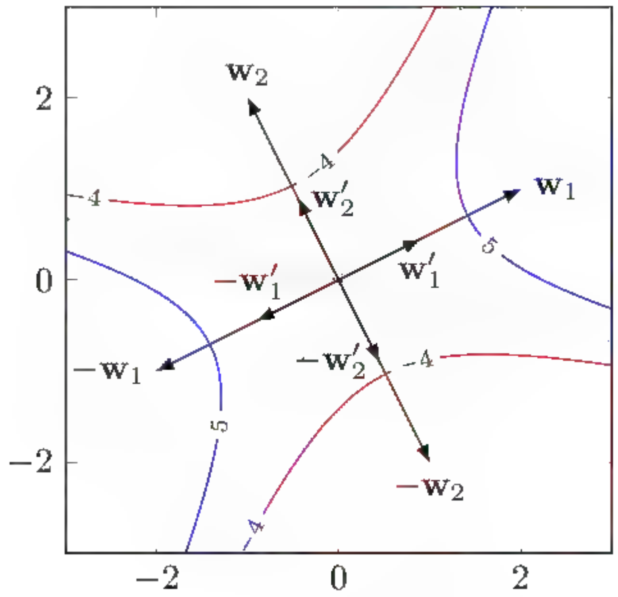


FIGURE 25.4.2. The level sets $x^2 + 4xy - 2y^2 = 5$ (blue) and $x^2 + 4xy - 2y^2 = -4$ (red).

As we have discussed, in (t_1', t_2')-coordinates the quadratic form $q_A(x, y) = x^2 + 4xy - 2y^2$ becomes $2{t_1'}^2 - 3{t_2'}^2$, so the level set $x^2 + 4xy - 2y^2 = 5$ is equivalent to

$$2{t_1'}^2 - 3{t_2'}^2 = 5;$$

this is a hyperbola that meets the "t_1'-axis" (i.e., the line $t_2' = 0$, which is $\mathrm{span}(\mathbf{w}_1')$) at the points $(t_1', t_2') = (\pm\sqrt{5/2}, 0)$ (i.e., $\pm\sqrt{5/2}\,\mathbf{w}_1'$) with $\pm\sqrt{5/2}$ between ± 1 and ± 2 (since $5/2$ is between 1 and $4 = 2^2$). It does not meet the t_2'-axis (the line $t_1' = 0$, which is $\mathrm{span}(\mathbf{w}_2')$) since $-3{t_2'}^2 = 5$ has no solution in $\mathbf{R}$.

Arguing similarly to what we did in Example 25.4.1, the asymptotes are $t_2' = \pm\sqrt{2/3}\,t_1'$ with "slopes" $\pm\sqrt{2/3}$ *after tilting one's head* to make the t_1'-axis horizontal and the t_2'-axis vertical. Since these slopes are < 1 in absolute value (as $2/3 < 1$), the asymptotes are nearer to the t_1'-axis than to the t_2'-axis. In Figure 25.4.2 the level curve $q_A = 5$ is drawn in blue with asymptotes as the dotted lines, nearer to the t_1'-axis through $\mathbf{w}_1'$ and $\mathbf{w}_1$ than to the t_2'-axis through $\mathbf{w}_2'$ and $\mathbf{w}_2$.

When the equation $x^2 + 4xy - 2y^2 = -4$ is written in terms of (t_1', t_2')-coordinates it becomes

$$2{t_1'}^2 - 3{t_2'}^2 = -4.$$

This is a hyperbola that meets the t_2'-axis (i.e., the line $t_1' = 0$, which is $\mathrm{span}(\mathbf{w}_2')$) at the points $(t_1', t_2') = (0, \pm\sqrt{4/3})$ (i.e., $\pm\sqrt{4/3}\,\mathbf{w}_2'$). The two branches approach the asymptotes $t_1' = \pm\sqrt{3/2}\,t_2'$, which are exactly the same as the lines $t_2' = \pm\sqrt{2/3}\,t_1'$ that were the asymptotes we computed for the level set $q_A(x, y) = 5$. The hyperbola $q_A = -4$ is sketched in red in Figure 25.4.2.

There is nothing special about the values $c = 5$ and $c = -4$ considered above. What really matters in such cases for which the *eigenvalues have opposite signs* (as above) is that: (i) the equation written in the reference frame of orthonormal eigenvectors for A has no cross-term, and (ii) the asymptotes are "nearer" to the line spanned by the eigenvector whose eigenvalue is *smaller* in absolute value (as can be worked out by computing the asymptotes in a typical example, such as $2{t_1'}^2 - {t_2'}^2 = c$, to remind oneself what the pattern is without having to memorize anything). ■

Example 25.4.3. Now we turn to a definite quadratic form $q(x, y)$, the level sets of which are always (possibly tilted) ellipses centered at the origin. Consider the quadratic form $q_A(x, y) = 13x^2 - 6xy + 5y^2$ associated with $A = \begin{bmatrix} 13 & -3 \\ -3 & 5 \end{bmatrix}$. This matrix has trace $13+5 = 18$ and determinant $(13)(5)-(-3)(-3) = 65 - 9 = 56 > 0$, so it is definite and its characteristic polynomial is $P_A(\lambda) = \lambda^2 - 18\lambda + 56 = (\lambda - 14)(\lambda - 4)$. (You can use the quadratic formula to figure out that the roots are 4 and 14 if you don't eye-ball the factorization.)

To find corresponding eigenvectors we want to find nonzero solutions to each of $A\mathbf{x} = 14\mathbf{x}$ and $A\mathbf{x} = 4\mathbf{x}$, by writing each as a pair of 2 equations in 2 unknowns. For $A\mathbf{x} = 14\mathbf{x}$ we get the eigenvector $\mathbf{w}_1 = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$ for the eigenvalue $\lambda_1 = 14$, and for $A\mathbf{x} = 4\mathbf{x}$ we get the eigenvector $\mathbf{w}_2 = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$ for the eigenvalue $\lambda_2 = 4$. Let $\mathbf{w}_i' = \mathbf{w}_i/\|\mathbf{w}_i\|$ be the associated unit eigenvectors. Writing vectors in terms of $\{\mathbf{w}_1', \mathbf{w}_2'\}$ as $t_1'\mathbf{w}_1' + t_2'\mathbf{w}_2'$, the diagonalization formula (24.2.2) yields

$$q_A(t_1'\mathbf{w}_1' + t_2'\mathbf{w}_2') = \lambda_1 {t_1'}^2 + \lambda_2 {t_2'}^2 = 14{t_1'}^2 + 4{t_2'}^2.$$

The level sets $q_A(x, y) = c$ written in (t_1', t_2')-coordinates therefore have the form $14{t_1'}^2 + 4{t_2'}^2 = c$, which are ellipses (for $c > 0$). These ellipses are tilted relative to the x-axis and y-axis, being aligned with the coordinate axes for the (t_1', t_2')-coordinate system, which is to say the lines $\mathrm{span}(\mathbf{w}_1') = \mathrm{span}(\mathbf{w}_1)$ and $\mathrm{span}(\mathbf{w}_2') = \mathrm{span}(\mathbf{w}_2)$. Since the positive coefficient of ${t_2'}^2$ is smaller $(4 < 14)$, the ellipse is *longer* along the t_2'-axis, which is to say it is longer along the span of $\mathbf{w}_2$. This is shown in Figure 25.4.3, which shows the orthogonal eigenvectors, the associated unit eigenvectors, and the ellipse level curves aligned along their directions.

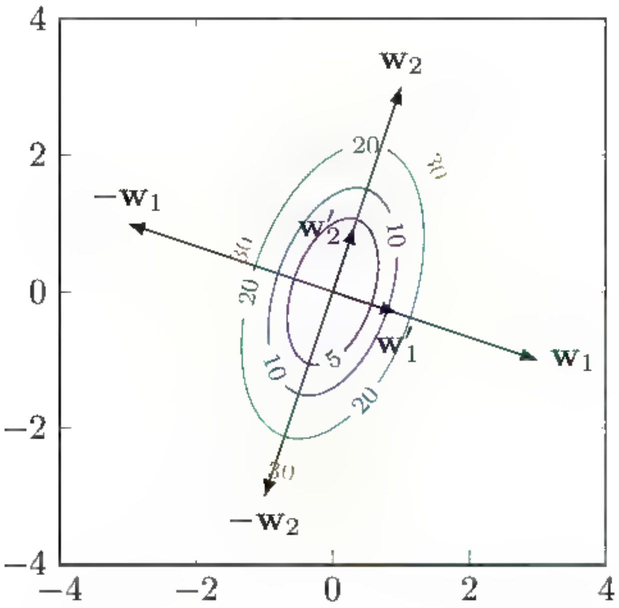


FIGURE 25.4.3. Some of the level sets of $13x^2 - 6xy + 5y^2$.

The general pattern in the definite case (so eigenvalues both positive, or both negative) goes exactly as in this example, by the same reasoning: the line spanned by the eigenvector whose eigenvalue has *smaller* absolute value is the direction along which the ellipse is *longer*. ■

Summary. Let's reformulate the discussion at the start of this section in light of the preceding eigenvector analysis applied to symmetric 2×2 Hessian matrices at critical points. If $f(x, y)$ has a critical point $\mathbf{a}$ at which the Hessian $(\mathrm{H}f)(\mathbf{a})$ has eigenvalues $\lambda_1, \lambda_2 \neq 0$ then in a **typically tilted reference frame** given by the orthogonal eigenlines the *quadratic approximation* to $f(\mathbf{x})$ near $\mathbf{a}$ has level sets

$$\lambda_1 {t_1'}^2 + \lambda_2 {t_2'}^2 = 2(c - f(\mathbf{a})). \tag{25.4.1}$$

These are good approximations to the level sets $f(\mathbf{x}) = c$ for $\mathbf{x}$ near $\mathbf{a}$ (up to tilting one's head), so:

(i) these level sets for f near $\mathbf{a}$ look like hyperbolas when λ_1, λ_2 have opposite signs (indefinite Hessian), with asymptotes nearer to the eigenline for $(\mathrm{H}f)(\mathbf{a})$ whose eigenvalue has *smaller* absolute value,

(ii) these level sets for f near $\mathbf{a}$ look like ellipses when λ_1, λ_2 have the same sign (definite Hessian), with the longer direction of the ellipse along the line spanned by the eigenvector whose eigenvalue has *smaller* absolute value.

In Chapter 26, algebraic considerations with eigenvalues will be combined with the preceding geometric lessons to conclude that when the eigenvalues λ_1, λ_2 are both nonzero then cases (i) and (ii) correspond *exactly* to two possibilities for the behavior of f near $\mathbf{a}$:

(i′) when the (nonzero) eigenvalues have opposite signs (the "hyperbola" case) then f has a saddle point at $\mathbf{a}$;

(ii′) when the (nonzero) eigenvalues have the same sign (the "ellipse" case) then f has a local extremum at $\mathbf{a}$.

In the latter case, we can distinguish local maxima from local minima either by inspecting numerically how the values of f near $\mathbf{a}$ behave – increasing or decreasing – as we approach $\mathbf{a}$ along a single direction, or (more conceptually) by computing the common sign of the eigenvalues of $(\mathrm{H}f)(\mathbf{a})$.

As will be explained in detail in Chapter 26 (with examples and an adaptation to the general multivariable case), the case of two positive eigenvalues corresponds to local minima and the case of two negative eigenvalues corresponds to local maxima (reminiscent of the sign cases for the single-variable second derivative test).

Since the quadratic approximation is a good approximation to $f(\mathbf{x})$ for $\mathbf{x}$ near $\mathbf{a}$, we conclude that the presence of such critical points and their local nature (saddle point, local maximum, or local minimum) can be determined by visually inspecting a contour plot to find nested collections of approximate ellipses around a point and nested collections of approximate hyperbolas around a point. Here is an example that illustrates all of these possibilities.

Example 25.4.4. For $f(x,y) = x^3 - 3(x^2+y^2) + y^3$, its surface graph is given in Figure 25.4.4.

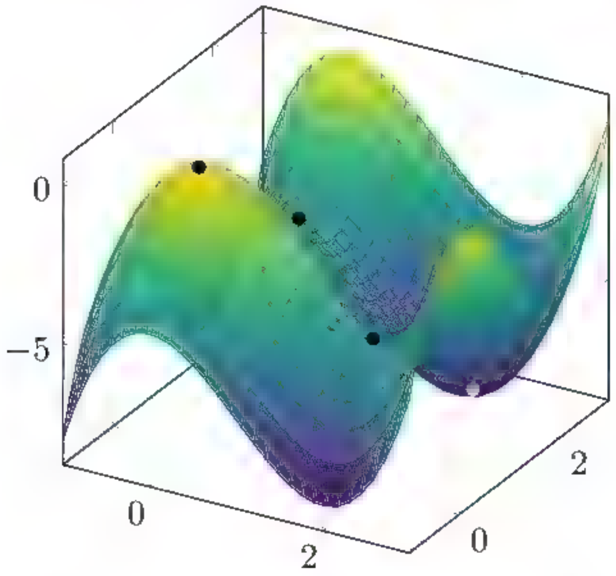


FIGURE 25.4.4. Graph of $f(x,y) = x^3 - 3(x^2+y^2) + y^3$, with dark dots at the local maximum $(0,0)$ and saddle points $(0,2)$ and $(2,0)$, and a yellow dot at the local minimum $(2,2)$.

This surface graph shows a local maximum, a local minimum, and two saddle points. We shall now recover that information by computing with gradients and Hessians, and interpret the outcome in terms of a contour plot.

The gradient of f is

$$\nabla f = \begin{bmatrix} 3x^2 - 6x \\ 3y^2 - 6y \end{bmatrix} = \begin{bmatrix} 3x(x-2) \\ 3y(y-2) \end{bmatrix},$$

so there are 4 critical points: $(0,0)$, $(0,2)$, $(2,0)$, $(2,2)$. Computing second partials, we get the general Hessian

$$\mathrm{H}f = \begin{bmatrix} 6x-6 & 0 \\ 0 & 6y-6 \end{bmatrix},$$

so at the 4 critical points the Hessian comes out as follows:

$$(\mathrm{H}f)(0,0) = \begin{bmatrix} -6 & 0 \\ 0 & -6 \end{bmatrix}, \quad (\mathrm{H}f)(2,2) = \begin{bmatrix} 6 & 0 \\ 0 & 6 \end{bmatrix}, \quad (\mathrm{H}f)(0,2) = \begin{bmatrix} -6 & 0 \\ 0 & 6 \end{bmatrix}, \quad (\mathrm{H}f)(2,0) = \begin{bmatrix} 6 & 0 \\ 0 & -6 \end{bmatrix}.$$

The corresponding quadratic forms $q_{(\mathrm{H}f)(\mathbf{a})}(\mathbf{h})$ for critical points $\mathbf{a}$ are

$$-(6h_1^2 + 6h_2^2), \;\; 6h_1^2 + 6h_2^2, \;\; -6h_1^2 + 6h_2^2, \;\; 6h_1^2 - 6h_2^2;$$

the corresponding quadratic approximations $f(\mathbf{a}+\mathbf{h}) \approx f(\mathbf{a}) + (1/2)q_{(\mathrm{H}f)(\mathbf{a})}(\mathbf{h})$ at critical points $\mathbf{a}$ are

$$f(h_1, h_2) \approx -(3h_1^2 + 3h_2^2), \;\; f(2+h_1, 2+h_2) \approx -8 + 3h_1^2 + 3h_2^2$$
$$f(h_1, 2+h_2) \approx -4 - 3h_1^2 + 3h_2^2, \;\; f(2+h_1, h_2) \approx -4 + 3h_1^2 - 3h_2^2$$

for (h_1, h_2) near $(0,0)$.

We conclude that the contour plot near $(0,0)$ and $(2,2)$ must consist of approximate ellipses since the eigenvalues have the same sign, and the contour plot near $(2,0)$ and $(0,2)$ must consist of approximate hyperbolas approaching an "X" since the eigenvalues have opposite signs. This is all encoded in the contour plot shown in Figure 25.4.5.

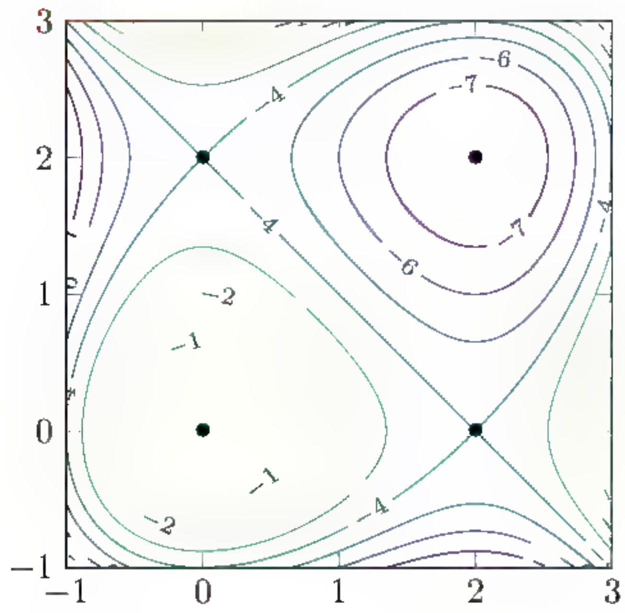


FIGURE 25.4.5. Contour plot of $f(x,y) = x^3 - 3(x^2+y^2) + y^3$. Note the four critical points.

Note that the eigenvalues for the Hessian at $(0,0)$ are negative and those for the Hessian at $(2,2)$ are positive. The surface graph in Figure 25.4.4 correspondingly exhibits a local maximum at $(0,0)$ and a local minimum at $(2,2)$, and saddle points at the other two critical points (whose Hessians we saw have eigenvalues with opposite signs). ■

25.5. Where does the quadratic approximation (25.3.1) come from? As a warm-up, let's consider the case $n = 1$, which is to say a function $f : \mathbf{R} \to \mathbf{R}$ as in single-variable calculus. Suppose we want to find *some* quadratic polynomial

$$Q(h) = A + Bh + Ch^2 \tag{25.5.1}$$

that is a good approximation to $f(a+h)$ when $h \approx 0$. How should we pick A, B, C?

A reasonable first guess is to try to match things when $h = 0$:

$$f(a+0) = Q(0), \text{ which says } f(a) = A + B(0) + C(0)^2 = A + 0 + 0 = A.$$

That leaves B and C undetermined. To get a better approximation, let's try to match first derivatives

$$f'(a+0) = Q'(0), \text{ which says } f'(a) = B + 2C(0) = B + 0 = B$$

(we have used that the derivative of $A + Bh + Ch^2$ with respect to h is $B + 2Ch$), and then second derivatives

$$f''(a+0) = Q''(0), \text{ which says } f''(a) = 2C$$

(we have used that the second derivative of $A + Bh + Ch^2$ with respect to h is $2C$).

So we have shown that $A = f(a)$, $B = f'(a)$, $C = f''(a)/2$. This gives the approximation

$$f(a) + f'(a)h + f''(a)\frac{h^2}{2}$$

for $f(a+h)$, recovering what one learns with Taylor series in single-variable calculus.

This idea will work for functions $f : \mathbf{R}^n \to \mathbf{R}$ with any n, to find a good quadratic approximation to f near a point $\mathbf{a} \in \mathbf{R}^n$. To make things more concrete, we treat the case of a function in two variables $f(x, y)$, and work out an approximation near a point (a, b).

A quadratic approximation would look like this:

$$f(a + h_1, b + h_2) \approx Q(h_1, h_2) = A + (B_1 h_1 + B_2 h_2) + (C_{11} h_1^2 + C_{12} h_1 h_2 + C_{22} h_2^2)$$

for some coefficients $A, B_1, B_2, C_1, C_{12}, C_{22}$ to be determined (compare this to (25.5.1)). We can write this in matrix language as follows:

$$Q(h_1, h_2) = A + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \cdot \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}^{\top} \begin{bmatrix} C_{11} & C_{12}/2 \\ C_{12}/2 & C_{22} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}. \tag{25.5.2}$$

(Please expand out the right side to see that it really does equal $Q(h_1, h_2)$; to have the symmetric matrix at the end with a common entry in the upper-right and lower-left, that entry must be half of the coefficient of $h_1 h_2$ in the definition of Q.)

As in the single-variable case, we will try to solve for all the unknowns on the right side – namely $A, B_1, B_2, C_{11}, C_{12}, C_{22}$– by differentiating with respect to h_1 and h_2 and then setting $h_1 = h_2 = 0$. That is, in order that $f(a + h_1, b + h_2) \approx Q(h_1, h_2)$ up to "second order" for all small h_1, h_2, we want to have:

$$\begin{gathered}
f(0,0) = Q(0,0), \\
\frac{\partial f}{\partial h_1}(a + h_1, b + h_2)\Big|_{(h_1,h_2)=(0,0)} = \frac{\partial Q}{\partial h_1}(0,0), \quad \frac{\partial f}{\partial h_2}(a + h_1, b + h_2)\Big|_{(h_1,h_2)=(0,0)} = \frac{\partial Q}{\partial h_2}(0,0), \\
\frac{\partial^2 f}{\partial h_1^2}(a + h_1, b + h_2)\Big|_{(h_1,h_2)=(0,0)} = \frac{\partial^2 Q}{\partial h_1^2}(0,0), \quad \frac{\partial^2 f}{\partial h_2^2}(a + h_1, b + h_2)\Big|_{(h_1,h_2)=(0,0)} = \frac{\partial^2 Q}{\partial h_2^2}(0,0), \\
\frac{\partial^2 f}{\partial h_1 \partial h_2}(a + h_1, b + h_2)\Big|_{(h_1,h_2)=(0,0)} = \frac{\partial^2 Q}{\partial h_1 \partial h_2}(0,0).
\end{gathered}$$

The evaluations of $(a + h_1, b + h_2)$ at $(h_1, h_2) = (0,0)$ are just evaluations at (a, b), and by change of dummy variable the values at $(h_1, h_2) = (0,0)$ of the partial derivatives of $f(a + h_1, b + h_2)$ with respect to h_1 and/or h_2 are the values at (a, b) of the partial derivatives of $f(x, y)$ with respect to x and/or y. In this way we can clean up the desired equalities a bit:

$$f(a,b) = Q(0,0), \quad f_x(a,b) = \frac{\partial Q}{\partial h_1}(0,0), \quad f_y(a,b) = \frac{\partial Q}{\partial h_2}(0,0), \tag{25.5.3}$$

$$f_{xx}(a,b) = \frac{\partial^2 Q}{\partial h_1^2}(0,0), \quad f_{yy}(a,b) = \frac{\partial^2 Q}{\partial h_2^2}(0,0), \quad f_{xy}(a,b) = \frac{\partial^2 Q}{\partial h_1 \partial h_2}(0,0). \tag{25.5.4}$$

But the right sides can be computed explicitly from the defining expression

$$Q(h_1, h_2) = A + (B_1h_1 + B_2h_2) + (C_{11}h_1^2 + C_{12}h_1h_2 + C_{22}h_2^2),$$

namely:

$$Q(0,0) = A, \quad \frac{\partial Q}{\partial h_1}Q(0,0) = B_1, \quad \frac{\partial Q}{\partial h_2}(0,0) = B_2,$$

$$\frac{\partial^2 Q}{\partial h_1^2}(0,0) = 2C_{11}, \quad \frac{\partial^2 Q}{\partial h_1 \partial h_2}(0,0) = C_{12}, \quad \frac{\partial^2 Q}{\partial h_2^2}(0,0) = 2C_{22}.$$

Plugging these into the expressions in (25.5.3) and (25.5.4), we obtain the constraints

$$f(a,b) = A, \quad f_x(a,b) = B_1, \quad f_y(a,b) = B_2, \quad f_{xx}(a,b) = 2C_{11}, \quad f_{xy}(a,b) = C_{12}, \quad f_{yy}(a,b) = 2C_{22}.$$

These express the unknown coefficients in terms of values of f and its partial derivatives up to second order:

$$A = f(a,b), \quad B_1 = f_x(a,b), \quad B_2 = f_y(a,b),$$

$$C_{11} = (1/2)f_{xx}(a,b), \quad C_{12} = f_{xy}(a,b), \quad C_{22} = (1/2)f_{yy}(a,b).$$

Plugging these into (25.5.2) yields exactly (25.3.1) in the case of a two-variable function $f(x,y)$; in particular, we have the factor $1/2$ throughout the 2×2 matrix in the quadratic part of the approximation (just as half of the Hessian shows up in (25.3.1)).

Chapter 25 highlights (links to highlights in [illegible] and next chapters)

Notation	Meaning	Location in text
$(\mathrm{H}f)(\mathbf{a})$	Hessian matrix for $f : \mathbf{R}^n \to \mathbf{R}$ at $\mathbf{a} \in \mathbf{R}^n$	Definition 25.2.1

Concept	Meaning	Location in text
Hessian matrix of $f : \mathbf{R}^n \to \mathbf{R}$	matrix whose ij-entry is $f_{x_i x_j}$	Definition 25.2.1
quadratic approximation to $f : \mathbf{R}^n \to \mathbf{R}$ near $\mathbf{a} \in \mathbf{R}^n$	refinement of gradient linear approximation, using $(1/2)q_{(\mathrm{H}f)(\mathbf{a})}(\mathbf{h})$ to get better approximation to $f(\mathbf{a}+\mathbf{h})$ for small $\mathbf{h}$	Definition 25.3.1

Result	Meaning	Location in text
symmetry of Hessian matrix	for $f : \mathbf{R}^n \to \mathbf{R}$ and $\mathbf{a} \in \mathbf{R}^n$, the matrix $(\mathrm{H}f)(\mathbf{a})$ is symmetric	Remark 25.2.2
for $n = 2$, eigenvector information for Hessian controls contour plot near **critical point** $\mathbf{a}$	if eigenvalues have same sign then plot near $\mathbf{a}$ is approximate nested ellipses aligned with the eigenvector directions; if eigenvalues have opposite signs then plot near $\mathbf{a}$ is approximate nested hyperbolas aligned with the eigenvector directions	Section 25.4 from (25.4.1) onwards

Skill	Location in text
compute Hessian of $f : \mathbf{R}^n \to \mathbf{R}$ symbolically and at specific $\mathbf{a} \in \mathbf{R}^n$	Example 25.2.4
compute quadratic approximation to $f : \mathbf{R}^n \to \mathbf{R}$ near $\mathbf{a} \in \mathbf{R}^n$	Examples 25.3.2, 25.3.3
for $n = 2$, use eigenvectors for symmetric 2×2 matrix A and the signs of the eigenvalues to sketch qualitatively correct level curves for $q_A(\mathbf{x})$ (nested ellipses or hyperbolas aligned with orthogonal lines spanned by eigenvectors: ellipse is longer in direction of such a line whose eigenvalue has *smaller* absolute value, hyperbola asymptotes are "closer" to such a line whose eigenvalue has *smaller* absolute value)	Examples 25.4.2, 25.4.3

26. Grand finale: application of the Hessian to local extrema, and bon voyage

In this chapter we use the quadratic approximation to a function $f : \mathbf{R}^n \to \mathbf{R}$ near a critical point $\mathbf{a}$ to formulate a multivariable second derivative test: to determine whether the graph of f inside $\mathbf{R}^{n+1}$ near $\mathbf{a}$ is either "bowl-shaped" downward (local maximum) or upward (local minimum) or perhaps looks more like a "saddle" or maybe is "too flat" to accurately describe using information from second partial derivatives.

The main tool is the symmetric Hessian matrix $(\mathrm{H}f)(\mathbf{a})$, or rather its associated quadratic form

$$\begin{aligned} q(\mathbf{x}) &= \sum_{i=1}^{n} b_{ii}x_i^2 \; + \; 2 \sum_{1\le i<j\le n} b_{ij}x_ix_j \\ &= (b_{11}x_1^2 + b_{22}x_2^2 + \cdots + b_{nn}x_n^2) + 2b_{12}x_1x_2 + 2b_{13}x_1x_3 + \cdots + 2b_{1n}x_1x_n \\ &\quad + 2b_{23}x_2x_3 + 2b_{24}x_2x_4 + \cdots + 2b_{2n}x_2x_n + 2b_{34}x_3x_4 + \cdots + 2b_{3n}x_3x_n + \cdots + 2b_{n-1,n}x_{n-1}x_n \end{aligned}$$

with $b_{ij} = f_{x_ix_j}(\mathbf{a})$, via the link in (20.3.1) and Example 20.3.12 between quadratic forms and symmetric matrices. (There are 2's in front of the cross-terms above because for $i < j$ the coefficient of x_ix_j in the quadratic form $q_M(\mathbf{x})$ associated with an $n \times n$ symmetric matrix M is $m_{ij} + m_{ji} = 2m_{ij}$; for $n = 2, 3$ see (20.3.2), (20.3.3) respectively. When $n = 2$ this is $q(x_1, x_2) = f_{x_1x_1}(\mathbf{a})x_1^2 + 2f_{x_1x_2}(\mathbf{a})x_1x_2 + f_{x_2x_2}(\mathbf{a})x_2^2$.)

A key question, seen early in Section 25.4, is: what is the geometry of the level sets $q(x_1, \dots, x_n) = c$? We will use **eigenvectors** to understand this in a systematic way for every n, building on what we saw in Section 25.4 for $n = 2$. Applying this to the quadratic form associated with the Hessian at a critical point $\mathbf{a}$ of a function $f : \mathbf{R}^n \to \mathbf{R}$ will tell us if $\mathbf{a}$ is a local maximum, local minimum, saddle point, or something else; for $n = 1$ this will recover the second derivative test from single-variable calculus.

In Section 26.5, we review your mathematical growth during this course and give advice on further courses in linear algebra.

By the end of this chapter, for $f : \mathbf{R}^n \to \mathbf{R}$ you should be able to:

- identify local maxima, local minima, and saddle points from a contour plot when $n = 2$;
- determine if a critical point $\mathbf{a} \in \mathbf{R}^n$ for f is a local maximum, local minimum, or a saddle point when $(\mathrm{H}f)(\mathbf{a})$ is a diagonal matrix;
- determine the definiteness of the Hessian in general using eigenvalues, and use that information at **critical points** to identify local maxima, local minima, and saddle points;
- know which Stanford Math course on further linear algebra is a better fit for your interests.

26.1. Definiteness and saddle points. To formulate a "second derivative test" for $f : \mathbf{R}^n \to \mathbf{R}$ with any n, first recall from Chapter 25 that if $f : \mathbf{R}^n \to \mathbf{R}$ is a function then the quadratic approximation to f near a point $\mathbf{a} \in \mathbf{R}^n$ is given by $f(\mathbf{a}+\mathbf{h}) \approx f(\mathbf{a}) + (\nabla f)(\mathbf{a}) \cdot \mathbf{h} + (1/2)\mathbf{h}^\top((\mathrm{H}f)(\mathbf{a}))\mathbf{h}$ for $\mathbf{h} \in \mathbf{R}^n$ near $\mathbf{0}$. For $n = 1$, this is the Taylor approximation $f(a + h) \approx f(a) + f'(a)h + (1/2)f''(a)h^2$ in single-variable calculus to second order in $h \in \mathbf{R}$ near 0.

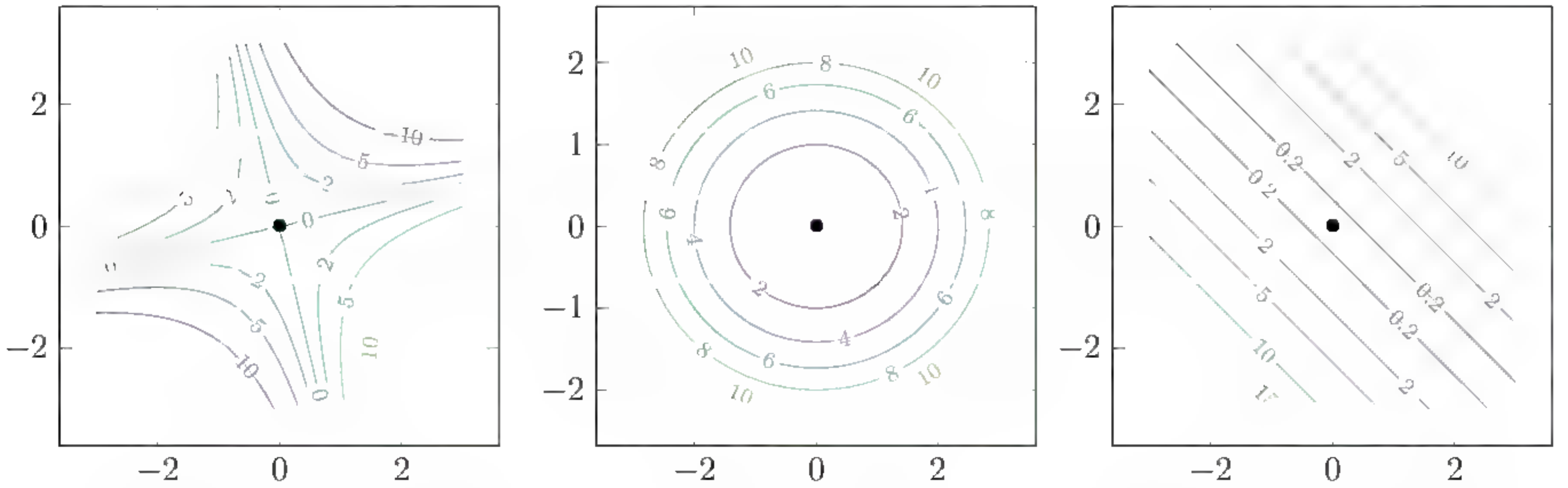
Suppose $\mathbf{a}$ is a critical point; i.e., $(\nabla f)(\mathbf{a}) = \mathbf{0}$. The quadratic approximation then becomes

$$f(\mathbf{a}+\mathbf{h}) \approx f(\mathbf{a}) + \frac{1}{2}\mathbf{h}^\top((\mathrm{H}f)(\mathbf{a}))\mathbf{h}. \tag{26.1.1}$$

Example 26.1.1. Consider $n = 1$, so this is the approximation $f(a+h) \approx f(a) + (1/2)f''(a)h^2$ at critical points in single-variable calculus. If $f''(a) > 0$ then the quantity $(1/2)f''(a)h^2$ is positive for all $h \neq 0$, so $f(a + h) > f(a)$ for all nonzero h near 0 – a local minimum at a. Similarly, if $f''(a) < 0$ then the quantity $(1/2)f''(a)h^2$ is negative for all small $h \neq 0$, so $f(a + h) < f(a)$ for all nonzero h near 0 – a local maximum at a. We have just *recovered the second derivative test in single-variable calculus* via the perspective of the quadratic approximation at a critical point, applied in the case $n = 1$. $\blacksquare$

To extend Example 26.1.1 to $n > 1$, by (26.1.1) we see $f(\mathbf{a}+\mathbf{h}) - f(\mathbf{a}) \approx (1/2)\mathbf{h}^\top((\mathrm{H}f)(\mathbf{a}))\mathbf{h}$ for all small $\mathbf{h}$. Hence, to determine if $f(\mathbf{a})$ is a local maximum (i.e., is $f(\mathbf{a}+\mathbf{h}) < f(\mathbf{a})$ for *all* small $\mathbf{h} \neq \mathbf{0}$) or a local minimum (i.e., is $f(\mathbf{a}+\mathbf{h}) > f(\mathbf{a})$ for *all* small $\mathbf{h} \neq \mathbf{0}$?), we want to know if $q_{(\mathrm{H}f)(\mathbf{a})}(\mathbf{h}) = \mathbf{h}^\top((\mathrm{H}f)(\mathbf{a}))\mathbf{h}$ is negative for *all* small $\mathbf{h}$ near $\mathbf{0}$ or positive for *all* small $\mathbf{h} \neq \mathbf{0}$.

In the terminology of Definition 24.2.2: is the symmetric Hessian matrix $(\mathrm{H}f)(\mathbf{a})$ (or equivalently $q_{(\mathrm{H}f)(\mathbf{a})}(\mathbf{x})$) negative-definite or positive-definite or something else? The possibilities for variation in the values of an n-variable quadratic form near $\mathbf{0}$ are *much* more complicated when $n > 1$ than when $n = 1$, as we see for $n = 2$ in contour plots near (0,0) for some quadratic forms $q(x_1, x_2)$ in Figure 26.1.1: the level sets for the first are hyperbolas, for the second are ellipses, and for the third are pairs of parallel lines.



Contour plot of $x_1^2 - 4x_1x_2 - x_2^2$. Contour plot of $x_1^2 + 2x_2^2$. Contour plot of $x_1^2 + 2x_1x_2 + x_2^2$.

FIGURE 26.1.1. Contour plots of several quadratic functions of 2 variables

For an $n \times n$ symmetric matrix A (such as $A = (\mathrm{H}f)(\mathbf{a})$ as above), let's think about the scalar $q_A(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x} \in \mathbf{R}$ as a function of $\mathbf{x} \in \mathbf{R}^n$. We have encountered functions of this type earlier, in Section 20.3, where we found for $n = 2, 3$:

$$\begin{bmatrix} x \\ y \end{bmatrix}^\top \begin{bmatrix} a & u \\ u & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & u \\ u & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = ax^2 + by^2 + 2uxy,$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}^\top \begin{bmatrix} a & u & v \\ u & b & w \\ v & w & c \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} a & u & v \\ u & b & w \\ v & w & c \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = ax^2 + by^2 + cz^2 + 2uxy + 2vxz + 2wyz.$$

(The contributions on the right side from the off-diagonal matrix entries all occur with a 2 in front.)

The quadratic approximation at critical points has motivated interest in determining where such functions $q(\mathbf{x})$ are positive or negative. When $n > 1$ there is an *additional* possibility with **no analogue** for $n = 1$, as we now discuss. First consider the case $n = 2$. For a critical point $\mathbf{a} = (a_1, a_2) \in \mathbf{R}^2$ of f, it might happen that the part of the surface graph $z = f(x, y)$ over some line L in $\mathbf{R}^2$ through $\mathbf{a}$ has a local minimum at $\mathbf{a}$ and the part of the surface graph $z = f(x, y)$ over another line L' in $\mathbf{R}^2$ through $\mathbf{a}$ has a local maximum at $\mathbf{a}$. This is visualized by a "saddle" appearance of the surface graph $z = f(x, y)$ near $(a_1, a_2, f(a_1, a_2)) \in \mathbf{R}^3$:

Example 26.1.2. For Example 10.2.8 with $f(x, y) = x^2 - y^2$, the contour plot in Figure 10.2.2 consists of hyperbolas and the surface graph in Figure 10.2.3 is a "saddle" over $(0, 0) \in \mathbf{R}^2$. The cause of the saddle shape is that there are vertical planes P and P' that cut the surface graph S through $(0, 0, f(0, 0))$ along respective curves C and C' with *opposite* local extrema behavior at $(0, 0, f(0, 0))$. Indeed, the vertical plane $x = 0$ cuts S along the curve $\{(0, y, f(0, y)) : y \in \mathbf{R}\}$ that is the upside-down parabola graph of $f(0,y) = -y^2$ with a local *maximum* at $y = 0$, and the vertical plane $y = 0$ cuts S along the curve $\{(x, 0, f(x, 0)) : x \in \mathbf{R}\}$ that is the parabola graph of $f(x, 0) = x^2$ with a local *minimum* at $x = 0$.

In general, a vertical plane through $(0, 0, f(0, 0))$ amounts to choosing a line ℓ in the xy-plane through $(0, 0)$ (where the vertical plane cuts the xy-plane), and the curve along which that vertical plane cuts the surface graph is exactly the graph of $f(x, y)$ over the points in the line ℓ. There are *many lines* ℓ through $(0, 0)$ aside from the coordinate axes along which $f(x, y) = x^2 - y^2$ has a local extremum at $(0, 0)$. For instance: on $y = 5x$ and $y = -3x$ there a local maximum at $(0, 0)$ since $f(x, 5x) = -24x^2$ and $f(x, -3x) = -8x^2$, whereas on $y = x/2$ and $y = -(2/3)x$ there is a local minimum at $(0, 0)$ since $f(x, x/2) = (3/4)x^2$ and $f(x, -(2/3)x) = (5/9)x^2$.

As another example, look at Figure 26.1.2 for the surface graph of $f(x, y) = \sin(x)\sin(y)$ that has a "saddle" shape over (π, π). Near (π, π), the contour plot in Figure 26.1.3 looks like a tilted version of the collection of hyperbolas in Figure 10.2.2 (and likewise for the contour plot on the left in Figure 26.1.1). If we look at f over the lines $y = x$ and $y = -x$, we get opposite local extrema behavior over the origin in those respective lines: $f(x, x) = \sin^2(x)$ has a local minimum at $x = 0$ and $f(x, -x) = -\sin^2(x)$ has a local maximum at $x = 0$. More generally, *looking at the labels on the level curves in the contour plot* shows that the behavior of f on any line $y = cx$ with $c > 0$ has a local minimum at the origin and the behavior of f on any line $y = -cx$ has a local maximum at the origin.

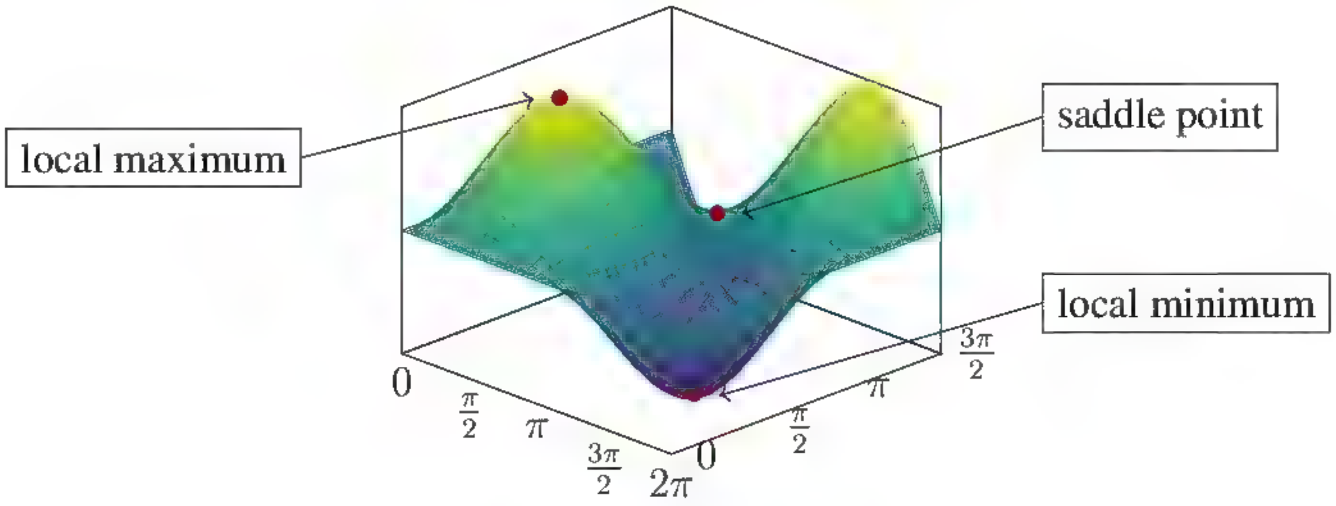


FIGURE 26.1.2. The graph of $f(x, y) = \sin(x)\sin(y)$

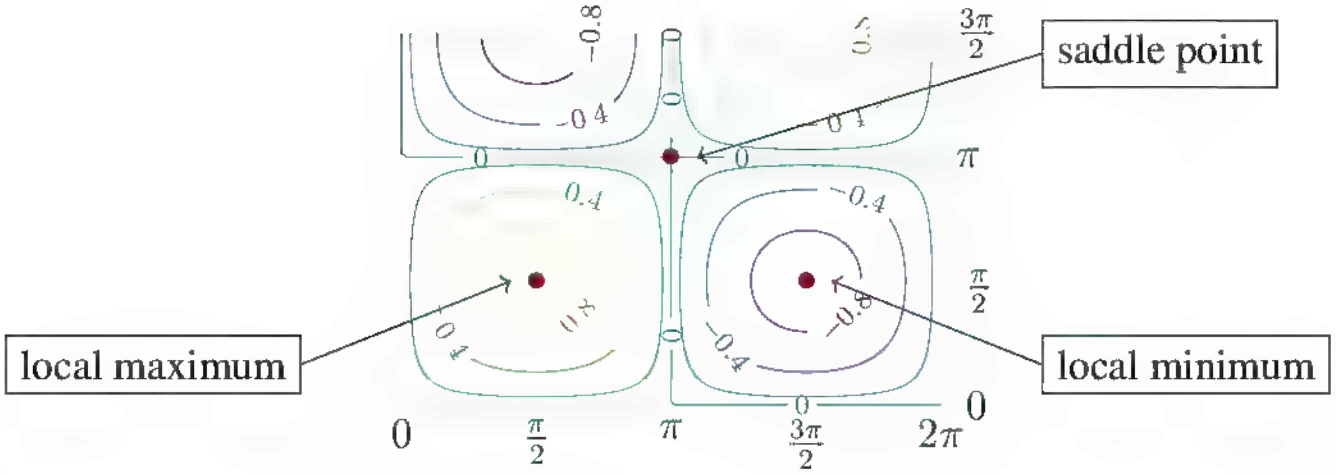


FIGURE 26.1.3. The contour plot of $f(x, y) = \sin(x)\sin(y)$. For $c > 0$, it meets $y = cx$ with a local minimum at the origin and meets $y = -cx$ with a local maximum at the origin.

■

The analysis in Example 26.1.2 inspires the terminology below (recalled from Section 10.2), but it will be the *geometry of the contour plot* and *not* of the surface graph that is the key to everything we will do in $\mathbf{R}^n$ for general $n > 1$.

Definition 26.1.3. For $n > 1$ and a function $f : \mathbf{R}^n \to \mathbf{R}$, a critical point $\mathbf{a} \in \mathbf{R}^n$ is called a *saddle point* of f if there are different lines L, L' in $\mathbf{R}^n$ through $\mathbf{a}$ so that f evaluated just on the line L has a local maximum at $\mathbf{a}$ and f evaluated just on the line L' has a local minimum at $\mathbf{a}$.

More explicitly, if we write L in a parametric form $\{\mathbf{a}+t\mathbf{v}\}$ and L' in a parametric form $\{\mathbf{a}+t\mathbf{v}'\}$ for nonzero vectors $\mathbf{v}$ and $\mathbf{v}'$ (these are not scalar multiples of each other, since L and L' are different lines through the common point $\mathbf{a}$) then the functions

$$g(t) = f(\mathbf{a}+t\mathbf{v}), \quad h(t) = f(\mathbf{a}+t\mathbf{v}')$$

encode the respective behavior of f on points of L and L'. The saddle-point condition is saying that g has a local maximum at $t = 0$ and h has a local minimum at $t = 0$.

Example 26.1.2 illustrates *tremendous flexibility* for lines L and L' through $\mathbf{a}$ that can work in Definition 26.1.3. However, the geometry of contour plot near a saddle point has a special feature illustrated for $n = 2$ in Figure 26.1.4: among pairs of lines through $\mathbf{a}$ with opposite local extrema behavior for f, there is a special pair (L and L' in Figure 26.1.4): the "lines of symmetry" for the contour plot near $\mathbf{a}$ (reflecting the plot across either line yields the original plot again, to good approximation). In Section 26.3 we'll see such lines are $\{\mathbf{a}+t\mathbf{w}\}$ and $\{\mathbf{a}+t\mathbf{w}'\}$ for orthogonal *eigenvectors* $\mathbf{w}, \mathbf{w}'$ of $(\mathrm{H}f)(\mathbf{a})$ (see Figure 26.3.1).

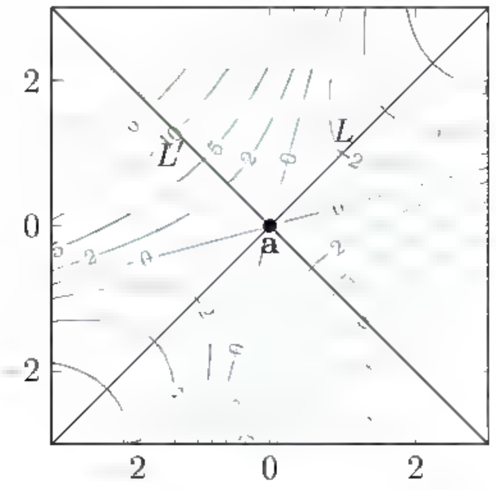
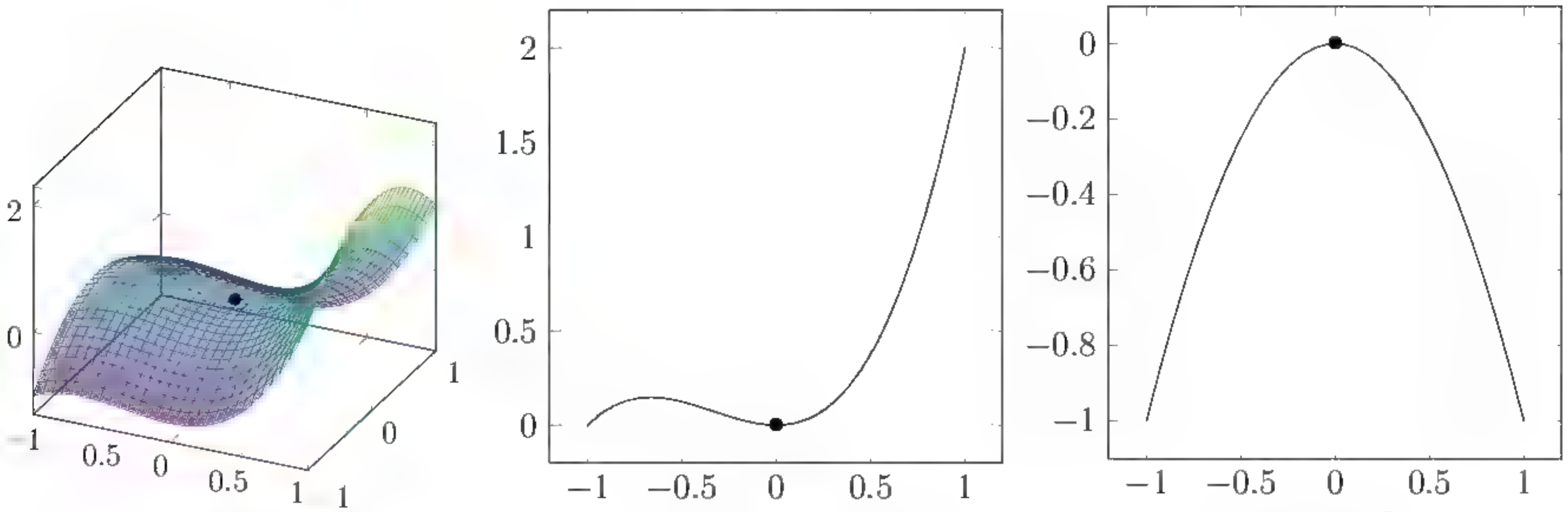


FIGURE 26.1.4. A saddle point at $\mathbf{a}$, along with a very special pair of lines L, L' through it.



$z = x^3 + x^2 - y^2$ $\quad$ $z = x^3 + x^2$ $\quad$ $z = -y^2$

FIGURE 26.1.5. At a saddle point, a local minimum over $y = 0$ and local maximum over $x = 0$.

Remark 26.1.4. Beware that some lines through a saddle point $\mathbf{a}$ may *fail to work either way* for a local extremum. Consider $f(x, y) = x^3 + x^2 - y^2$ as in Figure 26.1.5. This has a saddle point at $(0, 0)$ because its restrictions to the x-axis and y-axis are the respective functions $x^3 + x^2$ and $-y^2$ which respectively have a local minimum and a local maximum at the origin. But the restriction of f to the line $x = y$ through $(0, 0)$ is $y^3 + y^2 - y^2 = y^3$, which does *not* have a local extremum at $y = 0$.

Let's come back to the multivariable considerations immediately following the discussion for $n = 1$ in Example 26.1.1. By (26.1.1), if $\mathbf{a} \in \mathbf{R}^n$ is a critical point of f then for small $\mathbf{h}$ we have

$$f(\mathbf{a}+\mathbf{h}) \approx f(\mathbf{a}) + \frac{1}{2}\mathbf{h}^\top(\mathrm{H}f)(\mathbf{a})\mathbf{h} = f(\mathbf{a}) + \frac{1}{2}q_{(\mathrm{H}f)(\mathbf{a})}(\mathbf{h}) \tag{26.1.2}$$

(see (20.3.1) with $A = (\mathrm{H}f)(\mathbf{a})$), so for small $\mathbf{h}$ if $q_{(\mathrm{H}f)(\mathbf{a})}(\mathbf{h}) > 0$ then $f(\mathbf{a}+\mathbf{h}) > f(\mathbf{a})$ whereas if $q_{(\mathrm{H}f)(\mathbf{a})}(\mathbf{h}) < 0$ then $f(\mathbf{a}+\mathbf{h}) < f(\mathbf{a})$. Using how we *defined* the notions of positive-definite, negative-definite, and indefinite for symmetric matrices A (in terms of $q_A(\mathbf{x})$) in Definition 24.2.2, this yields:

Theorem 26.1.5 (Second Derivative Test, Version I). Let $f : \mathbf{R}^n \to \mathbf{R}$ be a function, and $\mathbf{a} \in \mathbf{R}^n$ a **critical point** of f.

(i) If the Hessian $(\mathrm{H}f)(\mathbf{a})$ is positive-definite then $\mathbf{a}$ is a local minimum of f.
(ii) If the Hessian $(\mathrm{H}f)(\mathbf{a})$ is negative-definite then $\mathbf{a}$ is a local maximum of f.
(iii) If the Hessian $(\mathrm{H}f)(\mathbf{a})$ is indefinite then $\mathbf{a}$ is a saddle point of f and is neither a local minimum nor a local maximum of f.

If the Hessian $(\mathrm{H}f)(\mathbf{a})$ is in none of these cases (i.e., positive-semidefinite or negative-semidefinite but *not* definite) then we need more information, just as the single-variable second derivative test is inconclusive when the second derivative vanishes.

(For $n = 1$, so $\mathbf{a}$ corresponds to a scalar $a \in \mathbf{R}$, (i) and (ii) together recover the single-variable second derivative test since $\mathbf{h}^\top(\mathrm{H}f)(\mathbf{a})\mathbf{h} = f''(a)h^2$ with $h \in \mathbf{R}$ the unique entry in the 1-vector $\mathbf{h}$).

To apply Theorem 26.1.5 we need to figure out the definiteness properties of the Hessian at $\mathbf{a}$. (Is it positive-definite? Negative-definite? Indefinite? Something else?) How is this done? For $n = 1$, this is the question in single-variable calculus as to whether $f''(a)$ is positive or negative (or 0). For $n > 1$, this task is much more challenging. **We will figure out definiteness via eigenvalues** (in Section 26.3). Also, *the indefinite case has practical applications*: see Appendix J for an application to molecular structure.

Remark 26.1.6. In Theorem 26.1.5, $\mathbf{a}$ is a critical point! It can be shown that $(\mathrm{H}f)(\mathbf{a})$ is positive-definite for *all* $\mathbf{a} \in \mathbf{R}^n$ precisely when f is "strictly convex": $f((1-t)\mathbf{v}+t\mathbf{w}) < (1-t)f(\mathbf{v}) + tf(\mathbf{w})$ for all $\mathbf{v} \neq \mathbf{w}$ and $0 < t < 1$. (The key is that $h(t) = f((1-t)\mathbf{v} + t\mathbf{w})$ satisfies $h''(t) = q_{(\mathrm{H}f)((1-t)\mathbf{v}+t\mathbf{w})}(\mathbf{w}-\mathbf{v})$, and functions $g(t)$ are strictly convex precisely when $g''(t)$ is positive everywhere.) Such functions are relevant in economics and thermodynamics, because a critical point is then a unique global extremum! The 2nd Law of Thermodynamics ensures that entropy on the "state space" of a molecular system has negative-definite Hessian at all reasonable equilibrium states.

If the symmetric $n \times n$ matrix $(\mathrm{H}f)(\mathbf{a})$ is diagonal then we can read off its definiteness properties by inspection of the signs of its diagonal entries, as the following examples show.

Example 26.1.7. Consider $\mathrm{H} = \begin{bmatrix} 11 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 4 \end{bmatrix}$. The diagonal entries are the coefficients in the associated quadratic form: $q_{\mathrm{H}}(x, y, z) = \begin{bmatrix} x \\ y \\ z \end{bmatrix}^\top \mathrm{H} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = 11x^2 + 3y^2 + 4z^2$. This is positive when $(x, y, z) \neq \mathbf{0}$. ■

For a general diagonal $\mathrm{H} = \begin{bmatrix} a_1 & 0 & 0 \\ 0 & a_2 & 0 \\ 0 & 0 & a_3 \end{bmatrix}$ we have $q_{\mathrm{H}}(x,y,z) = \begin{bmatrix} x \\ y \\ z \end{bmatrix}^\top \mathrm{H} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = a_1x^2 + a_2y^2 + a_3z^2$. If all a_j's are positive then $q_{\mathrm{H}}(x,y,z) > 0$ whenever $(x,y,z) \neq \mathbf{0}$ (i.e., at least one of x, y, z is nonzero), so H is positive-definite if $a_1 > 0, a_2 > 0, a_3 > 0$. Likewise, H is indefinite if one of the a_i's is positive and another is negative, as in the third case of the following example.

Example 26.1.8.

$\begin{bmatrix} 11 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{bmatrix}$ is positive-definite, $\begin{bmatrix} -3 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -3 \end{bmatrix}$ is negative-definite, and $\begin{bmatrix} 3 & 0 & 0 \\ 0 & -4 & 0 \\ 0 & 0 & -11 \end{bmatrix}$ is indefinite. ■

The lesson is that it is easy to check definiteness or indefiniteness of any diagonal $n \times n$ matrix D by inspection (corresponding to evaluating $q_D(\mathbf{x})$ on the coordinate axes). This is applied to some Hessians in Section 26.2. There are two special non-diagonal cases whose definiteness also can be understood "by inspection", as we illustrate with the next two examples.

Example 26.1.9. Suppose that H is a 2×2 matrix that is "anti-diagonal", meaning that it looks like this: $\mathrm{H} = \begin{bmatrix} 0 & 5 \\ 5 & 0 \end{bmatrix}$. This is indefinite because $\begin{bmatrix} x \\ y \end{bmatrix}^\top \mathrm{H} \begin{bmatrix} x \\ y \end{bmatrix} = 10xy$ has positive and negative values (plug in (x,y) with various signs to see why). The same happens for any nonzero symmetric 2×2 anti-diagonal matrix (corresponding to $q(x,y) = cxy$ with $c \neq 0$). ■

Example 26.1.10. If M is any $m \times n$ matrix, the $n \times n$ matrix $B = M^\top M$ (called a *Gram* matrix) is symmetric (see Theorem 20.3.8 and especially Remark 20.3.9) and rarely diagonal. Some matrix algebra yields more: for any $\mathbf{v} \in \mathbf{R}^n$ we have

$$q_B(\mathbf{v}) = \mathbf{v}^\top(M^\top M)\mathbf{v} = (\mathbf{v}^\top M^\top)(M\mathbf{v}) = (M\mathbf{v})^\top(M\mathbf{v}) = (M\mathbf{v}) \cdot (M\mathbf{v}) = \|M\mathbf{v}\|^2 \geq 0.$$

Therefore B is *positive-semidefinite*, and if the null space of M is $\{\mathbf{0}\}$ then B is positive-definite. (In Examples 20.3.14 and 24.2.7 we mentioned some square matrices from materials science that are symmetric and positive-semidefinite. These properties can also be explained by physical arguments which show that those matrices have the form $M^\top M$.) Among the many contexts where Gram matrices arise are statistical modeling, quantum chemistry, computational biology, and kernel methods in machine learning. The killer app of Gram matrices is that for any rectangular matrix M of data, to apply any modern data science to M the first step is to compute the "singular value decomposition" (SVD) of M (to be explained in Section 27.3) that is just a re-interpretation of the Spectral Theorem for the Gram matrix $M^\top M$.

Amazingly, a reverse result holds: *every* positive-semidefinite symmetric $n \times n$ matrix B is $M^\top M$ for an $n \times n$ matrix M (so sometimes "positive-semidefinite symmetric" – called "Mercer's condition" in machine learning – is taken as the *definition* of Gram matrix). Indeed, by Theorem 24.4.1 we have $B = WDW^\top$ for orthogonal W and diagonal D whose entries are the eigenvalues λ_i of B, and all $\lambda_i \geq 0$ since B is positive-semidefinite (this is a variant of Proposition 24.2.10(i), with the same proof). Thus, if D' is the diagonal matrix with ii-entry $\sqrt{\lambda_i} \geq 0$ then $D = (D')^2 = D'D'^\top$, so $B = WD'D'^\top W^\top = (WD')(WD')^\top$. Hence, $B = M^\top M$ for $M = (WD')^\top$. We can do better (when B is nonzero, so M is nonzero)! The QR-decomposition (as in Section 22.5) for the possibly non-invertible M is QR where Q is an $n \times k$ matrix with orthonormal columns (with $k = \dim C(M)$) and R is a $k \times n$ upper triangular matrix with positive diagonal. Since $Q^\top Q = \mathrm{I}_k$, we have $B = M^\top M = (QR)^\top(QR) = (R^\top Q^\top)QR = R^\top(Q^\top Q)R = R^\top \mathrm{I}_k R = R^\top R$.

This special LU-decomposition ($L = U^\top$, with $U = R$ having positive diagonal) is called the *Cholesky decomposition*, discovered 14 years before the general LU-decomposition. It is useful for: inverting both covariance matrices in least squares and Hessians in Newton's method for optimization (Appendix I), simulations of correlated returns in quantitative finance [AI, I.5.7.4-I.5.7.5], and "square root filtering" to efficiently overcome noise in computer-based navigation (for GPS, space travel, etc.)

■

26.2. Worked examples of Version I of the Second Derivative Test (Theorem 26.1.5).

Example 26.2.1. Consider $h(x, y) = x \sin x - \cos y$. We encountered h in Example 10.2.9, where we saw $h_x = \sin x + x \cos x$ and $h_y = \sin y$, so $(0, 0)$ is a critical point. The restrictions $h(x, 0) = x \sin x - 1$ and $h(0, y) = -\cos y$ of h to the coordinate axes have a local minimum at the origin of the axes, but that doesn't ensure that $h(x, y)$ has a local minimum at $(0, 0)$: being a local extremum at a point in $\mathbf{R}^2$ is much stronger than being a local extremum on two lines passing through the point.

But with the multivariable second derivative test now in our toolkit, we can analyze the question of whether $(0, 0)$ is a local minimum for h. The Hessian matrix for h is diagonal everywhere: it is

$$\begin{bmatrix} h_{xx} & h_{yx} \\ h_{xy} & h_{yy} \end{bmatrix} = \begin{bmatrix} 2\cos x - x \sin x & 0 \\ 0 & \cos y \end{bmatrix}.$$

At $(0, 0)$ this is $\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$, which is positive-definite since its associated quadratic form is $2x^2 + y^2$. Hence, by Theorem 26.1.5, the critical point $(0, 0)$ for h is a local minimum for h (illustrated in Figure 26.2.1).

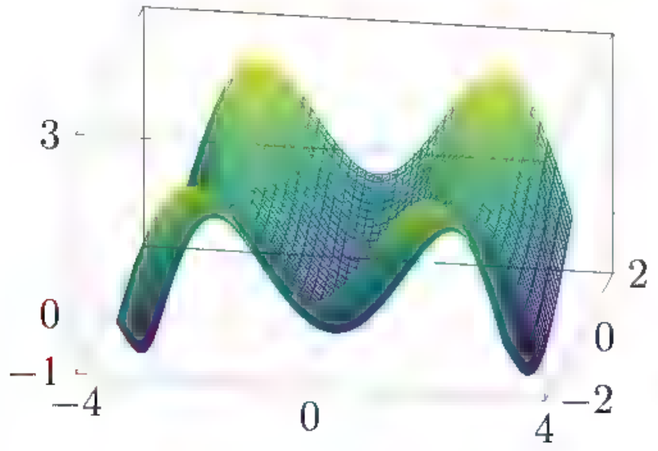


FIGURE 26.2.1. The surface graph of $h(x, y) = x \sin x - \cos y$

■

Remark 26.2.2. We shall carry out most examples below with 2-variable functions so you can see that the Hessian is encoding the geometry of both the contour plot and surface graph at each critical point. But the real significance is for applications beyond the 2-variable case, where the contour plot and surface graph are no longer available yet we need a mechanism to identify local maxima and local minima. The application of the Hessian in chemistry in Appendix J is an n-variable situation with $n > 3$.

Example 26.2.3. Let's find all the critical points of $g(x, y) = 3x^2 y + 2y^3 - xy$ and classify each as a local maximum, local minimum, or a saddle point. This example will have the "lucky" feature that the definiteness properties of the symmetric 2×2 Hessian matrix at every critical point can be read off by inspection – it turns out to be diagonal or anti-diagonal at each such point – even though its Hessian at most points is neither diagonal nor anti-diagonal. Generally one can't expect to be so lucky.

First of all, $(\nabla g)(x, y) = (6xy - y, 3x^2 + 6y^2 - x)$. For this to be $\mathbf{0}$, we must have

$$6xy - y = 0, \quad 3x^2 + 6y^2 = x.$$

The first condition says $y(6x-1)=0$, so either $y=0$ (in which case $3x^2=x$, so $x=0$ or $x=1/3$) or $x=1/6$ (in which case $6y^2=1/12$, so $y=\pm 1/(6\sqrt{2})$). Hence, the critical points are

$$P=(0,0), Q=(1/3,0), R=(1/6,1/(6\sqrt{2})), S=(1/6,-1/(6\sqrt{2})),$$

all shown in the contour plot in Figure 26.2.2.

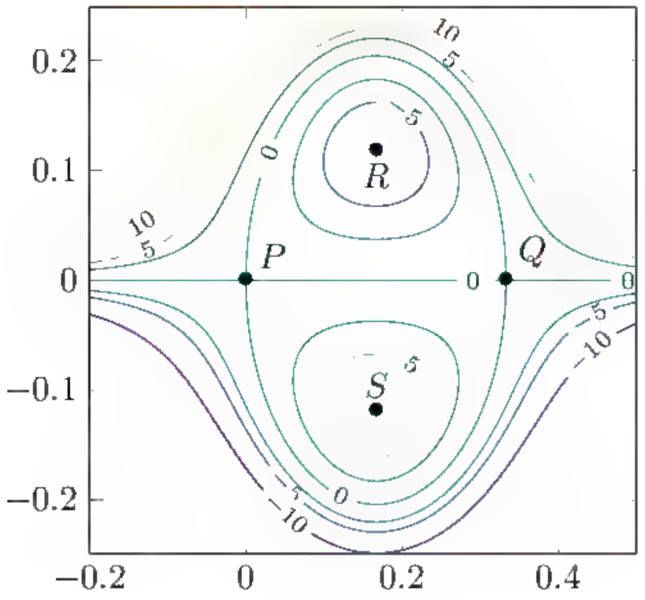


FIGURE 26.2.2. Contour plot of $g(x,y)=3x^2y+2y^3-xy$. The contour labels have been rescaled by a factor of 10^3 (e.g., the contour labeled 5 is the level set $g(x,y)=5\times 10^{-3}$).

We now compute the Hessian: $g_{xx}=6y, g_{xy}=6x-1, g_{yy}=12y$, so $(\mathrm{H}g)(x,y)=\begin{bmatrix} 6y & 6x-1 \\ 6x-1 & 12y \end{bmatrix}$. Next, we plug the coordinates of each critical point into this formula for the Hessian, and can read off the desired conclusions because each Hessian turns out to be simple enough that we can eyeball the answer as in Examples 26.1.7 and 26.1.9.

- At $P=(0,0)$ the Hessian is $\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$. This is indefinite by the same reasoning as in Example 26.1.9, so $(0,0)$ is a saddle point.
- At $Q=(1/3,0)$ the Hessian is $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. This is again indefinite, so $(1/3,0)$ is a saddle point.
- At $R=(1/6,1/6\sqrt{2})$ the Hessian is $\begin{bmatrix} 1/\sqrt{2} & 0 \\ 0 & 2/\sqrt{2} \end{bmatrix}$. This is a diagonal matrix with positive entries, so positive-definite; thus $(1/6,1/(6\sqrt{2}))$ is a local minimum.
- At $S=(1/6,-1/6\sqrt{2})$ the Hessian is $\begin{bmatrix} -1/\sqrt{2} & 0 \\ 0 & -2/\sqrt{2} \end{bmatrix}$. This is a diagonal matrix with negative entries, so negative-definite; thus $(1/6,-1/(6\sqrt{2}))$ is a local maximum.

In Figure 26.2.2, at each critical point the definiteness property we read off from the Hessian matches the expected geometry of the contour plot as discussed in Remark 26.3.5. We could use the contour plot to read off the location and behavior of the critical points, but the linear algebra of the Hessian tells us everything about critical points *without* needing to look at a contour plot; the importance of this was discussed in Remark 26.2.2. ■

Example 26.2.4. In Example 12.4.1 we studied a *constrained* optimization problem typically arising out of thermodynamics which came down to seeking critical points of

$$F(p,q,r)=(-p\ln(p)-q\ln(q)-r\ln(r))-\lambda(p+q+r-1)-\lambda'(2p+3q+5r-4)$$

on the region $p, q, r > 0$. We found that there was exactly one such point, depending on λ and λ', and it satisfied the constraint for only one choice of (λ, λ'). But we did not address how to know that the point we found (approximately) is actually a maximum rather than a minimum, or maybe a saddle point or something worse. The physical motivation in thermodynamics had us seeking a maximum, so that provides some reassurance. As a systematic mathematical method which can be used even in contexts where one doesn't have guidance from other considerations, we now apply the Second Derivative Test from Theorem 26.1.5.

The second partials of $F(p, q, r)$ work out very nicely: the first partials were found in Example 12.4.1, and taking partial derivatives of those gives the Hessian as a diagonal matrix

$$(\mathrm{H}F)(p,q,r) = \begin{bmatrix} -1/p & 0 & 0 \\ 0 & -1/q & 0 \\ 0 & 0 & -1/r \end{bmatrix}$$

(not involving λ or λ' because the original constraints were linear in p, q, r). The diagonal entries are visibly always negative (since $p, q, r > 0$), so this is negative-definite for any (p, q, r). In particular, at the point we found in (12.4.3) the Hessian is negative-definite, so that point must be a *local* maximum.

To justify mathematically that this local maximum (subject to the given constraint) is a *global* maximum (subject to the given constraint), one needs to carry out an additional argument involving the behavior of F near the "boundary" given by the coordinate planes $p = 0$, $q = 0$, and $r = 0$. We omit this additional argument since it goes beyond the level of the course. ■

Example 26.2.5. In Example 12.3.1 (which was a setting typical for certain optimization problems in economics) we saw that for a constant $\lambda > 0$ the function $P(x, y) = x^{1/4}y^{1/2} - \lambda(x + y - 3/4)$ (with $x, y > 0$) has exactly one critical point:

$$(a, b) = (1/(64\lambda^4), 1/(32\lambda^4)).$$

Is this a local maximum, a local minimum, a saddle point, or perhaps something worse? And if a local extremum, is it a global extremum?

To figure out what is going on near this point, let's analyze the Hessian. First, we work out the first partial derivatives:

$$\frac{\partial P}{\partial x} = \frac{1}{4}x^{\ 3/4}y^{1/2} - \lambda, \quad \frac{\partial P}{\partial y} = \frac{1}{2}x^{1/4}y^{\ 1/2} - \lambda.$$

Passing to the second partials, the constant λ conveniently disappears and we obtain

$$\frac{\partial^2 P}{\partial x^2} = -\frac{3}{16}x^{\ 7/4}y^{1/2}, \quad \frac{\partial^2 P}{\partial x \partial y} = \frac{\partial^2 P}{\partial y \partial x} = \frac{1}{8}x^{\ 3/4}y^{\ 1/2}, \quad \frac{\partial^2 P}{\partial y^2} = -\frac{1}{4}x^{1/4}y^{\ 3/2}.$$

Hence, the Hessian matrix for P at any point is

$$(\mathrm{H}P)(x,y) = \begin{bmatrix} -\frac{3}{16}x^{\ 7/4}y^{1/2} & \frac{1}{8}x^{\ 3/4}y^{\ 1/2} \\ \frac{1}{8}x^{\ 3/4}y^{\ 1/2} & -\frac{1}{4}x^{1/4}y^{\ 3/2} \end{bmatrix}.$$

With a bit of care in the algebra, plugging in the coordinates of the unique critical point (a, b) yields

$$(\mathrm{H}P)(a,b) = \begin{bmatrix} -48\lambda^5 & 16\lambda^5 \\ 16\lambda^5 & -16\lambda^5 \end{bmatrix} = 16\lambda^5 \begin{bmatrix} -3 & 1 \\ 1 & -1 \end{bmatrix}. \tag{26.2.1}$$

This is not diagonal, so we don't see its definiteness properties so easily. The way out of this conundrum will be to use information in the eigenvalues of this matrix; we will come back to it in Example 26.4.3. ■

Typically the Hessian is *not* diagonal, and in Section 26.3 we will handle such cases by invoking the full power of the Spectral Theorem via the diagonalization formula (24.2.2) using eigenvectors (yielding a reformulation of Theorem 26.1.5, given in Theorem 26.3.1, that is better-suited to applications). In other words, we apply Proposition 24.2.10 to the symmetric $n \times n$ matrix $A = (\mathrm{H}f)(\mathbf{a})$. This will lead to a satisfactory way of classifying critical points, to be illustrated in Section 26.4.

In this course (and in particular on exams) we are not going to dwell on critical points whose Hessian is in *none* of the cases of being positive-definite, negative-definite, or indefinite. However, such "degenerate" Hessians are very useful in computer vision (as we'll discuss in Remark 26.4.6). For the sake of completeness, here is an explicit example of one of these other possibilities.

Example 26.2.6. Let $f(x,y) = (x^2+y^2-3x)^2+4x^2(x-2)$. For $\mathbf{a} = (0,0)$, the gradient $(\nabla f)(\mathbf{a})$ vanishes, so $\mathbf{a}$ is a critical point of f. The Hessian at this point is $(\mathrm{H}f)(\mathbf{a}) = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$, so $q_{(\mathrm{H}f)(\mathbf{a})}(t_1,t_2) = 2t_1^2$. This is positive-semidefinite, but it is *not* positive-definite since for any nonzero t_2 its value at $(0,t_2) \neq (0,0)$ vanishes. The contour plot of $f(x,y)$ near such a "degenerate" critical point, as well as the level curve $f=0$ through this point, are rather complicated: see Figure 26.2.3. ■

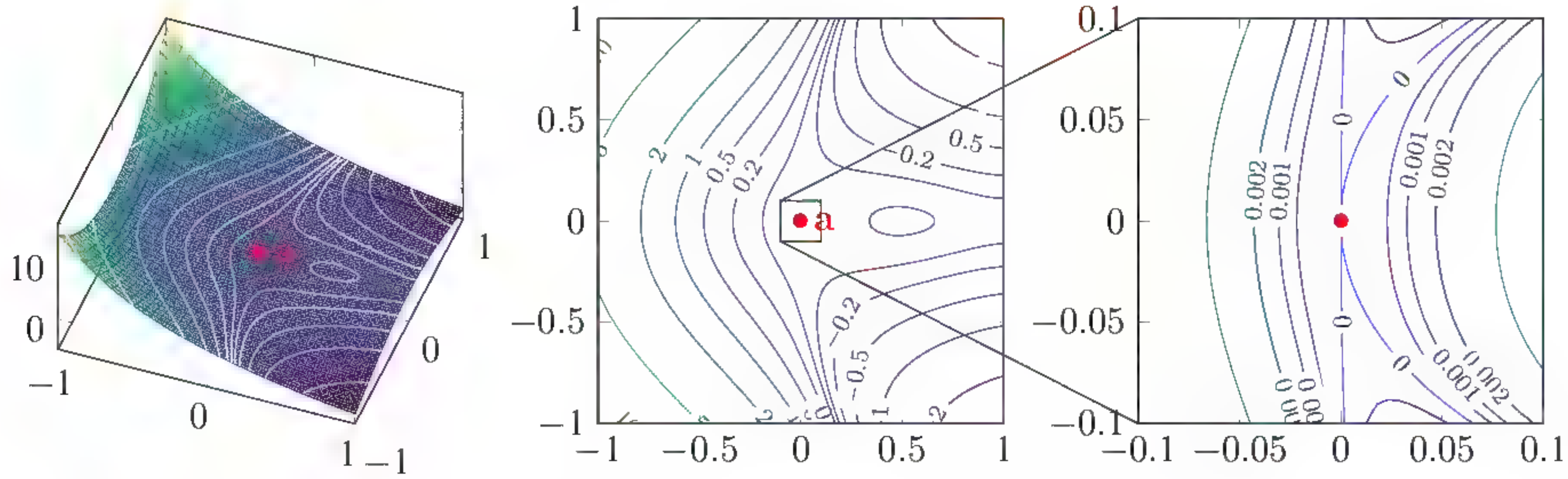


FIGURE 26.2.3. The function $f(x,y) = (x^2+y^2-3x)^2+4x^2(x-2)$ has a "degenerate" critical point at $\mathbf{a} = (0,0)$. Its surface graph is on the left, and its contour plot in the middle is zoomed in 10-fold on the right showing the blue level curve $f=0$ through $\mathbf{a}$ with a "tacnode" singularity there.

26.3. Eigenvectors and Hessians. For $f : \mathbf{R}^2 \to \mathbf{R}$, suppose the Hessian $(\mathrm{H}f)(\mathbf{a})$ at a critical point $\mathbf{a}$ has orthogonal eigenvectors $\mathbf{w}_1$ and $\mathbf{w}_2$ with respective eigenvalues 5 and -3. For scalars h_1, h_2 near 0, the quadratic approximation (26.1.1) applied to the vector $\mathbf{h} = h_1\mathbf{w}_1 + h_2\mathbf{w}_2$ near $\mathbf{0}$ yields

$$f(\mathbf{a}+h_1\mathbf{w}_1+h_2\mathbf{w}_2) \approx f(\mathbf{a}) + \frac{1}{2}q_{(\mathrm{H}f)(\mathbf{a})}(h_1\mathbf{w}_1+h_2\mathbf{w}_2) = f(\mathbf{a}) + \frac{5}{2}(\mathbf{w}_1\cdot\mathbf{w}_1)h_1^2 - \frac{3}{2}(\mathbf{w}_2\cdot\mathbf{w}_2)h_2^2,$$

the final equality using the diagonalization formula (24.2.2). Thus, on the line $L = \{\mathbf{a}+h_1\mathbf{w}_1\}$ (i.e., setting h_2 to be 0), the deviation of $f(\mathbf{a}+h_1\mathbf{w}_1)$ from $f(\mathbf{a})$ is $(5/2)(\mathbf{w}_1\cdot\mathbf{w}_1)h_1^2 > 0$ for small $h_1 \neq 0$, so $\mathbf{a}$ is a local minimum along L. On the line $L' = \{\mathbf{a}+h_2\mathbf{w}_2\}$ (i.e., setting h_1 to be 0), the deviation of $f(\mathbf{a}+h_2\mathbf{w}_2)$ from $f(\mathbf{a})$ is $-(3/2)(\mathbf{w}_2\cdot\mathbf{w}_2)h_2^2 < 0$ for small $h_2 \neq 0$, so $\mathbf{a}$ is a local maximum along L'.

If instead the eigenvalues are both positive, say 5 and 3, then the behavior of f near $\mathbf{a}$ would be governed by the quadratic approximation $f(\mathbf{a}+h_1\mathbf{w}_1+h_2\mathbf{w}_2) \approx f(\mathbf{a}) + \frac{5}{2}(\mathbf{w}_1\cdot\mathbf{w}_1)h_1^2 + \frac{3}{2}(\mathbf{w}_2\cdot\mathbf{w}_2)h_2^2$ for small h_1, h_2, so the value of f at nearby points is strictly bigger than $f(\mathbf{a})$ and hence f has a local minimum at $\mathbf{a}$. Likewise, if the eigenvalues are both negative, say -5 and -3, then for similar reasons f has a local maximum at $\mathbf{a}$. More generally, if $\mathbf{w}_i$ is an eigenvector for $(\mathrm{H}f)(\mathbf{a})$ with eigenvalue $\lambda_i \neq 0$ then (for h near 0) $f(\mathbf{a}+h\mathbf{w}_i) \approx f(\mathbf{a}) + (\lambda_i/2)(\mathbf{w}_i\cdot\mathbf{w}_i)h^2$. Thus, on the line through $\mathbf{a}$ in the directions of $\pm\mathbf{w}_i$ the function f deviates from $f(\mathbf{a})$ in a manner *controlled entirely by the sign* of λ_i (since $\mathbf{w}_i\cdot\mathbf{w}_i > 0$).

So if $(\mathrm{H}f)(\mathbf{a})$ has eigenvalues $\lambda > 0$ and $\lambda' < 0$ (i.e., opposite signs) then the corresponding eigenlines tell us $\mathbf{a}$ is a saddle point! In Figure 26.1.4, showing the contour plot of $x^2 - y^2$, the eigenlines for the Hessian at its critical point $(0,0)$ are the lines of symmetry L and L' shown there. In general, reasoning as above (i.e., applying Proposition 24.2.10 and the Spectral Theorem to the symmetric Hessian, and applying the diagonalization formula (24.2.2) to its associated quadratic form) yields the very applicable:

Theorem 26.3.1 (Second Derivative Test, Version II). Let $f : \mathbf{R}^n \to \mathbf{R}$ be a function, and $\mathbf{a} \in \mathbf{R}^n$ a **critical point** of f.

(i) If the Hessian $(\mathrm{H}f)(\mathbf{a})$ has all eigenvalues positive then $\mathbf{a}$ is a local minimum of f.
(ii) If the Hessian $(\mathrm{H}f)(\mathbf{a})$ has all eigenvalues negative then $\mathbf{a}$ is a local maximum of f.
(iii) If the Hessian $(\mathrm{H}f)(\mathbf{a})$ has both a positive eigenvalue and a negative eigenvalue (some other eigenvalue may be 0) then $\mathbf{a}$ is a saddle point of f and so it is not a local extremum of f.

(For $n = 1$, so the n-vector $\mathbf{a}$ is just a scalar $a \in \mathbf{R}$, (i) and (ii) together recover the second derivative test in single-variable calculus since $(\mathrm{H}f)(\mathbf{a})$ is then a 1×1 matrix with entry $f''(a)$.)

Remark 26.3.2. It may seem that Version II is a mere reformulation of Version I (as the diagonalization formula (24.2.2) directly relates (in)definiteness to signs of the eigenvalues), but that is wrong. Version I expresses the quadratic approximation with $(\mathrm{H}f)(\mathbf{a})$ at a critical point $\mathbf{a}$, but to arrive at Version II we need the *existence* of an orthogonal basis of eigenvectors for $(\mathrm{H}f)(\mathbf{a})$ (in order to apply the diagonalization formula!). The existence of such eigenvectors is the Spectral Theorem, so that is the bridge to Version II.

To apply Version II of the Second Derivative Test, we need a way to compute eigenvalues or at least their signs (without computing eigenvalues explicitly). For $n > 2$ there are ways to do this quickly on a computer (when n isn't too huge), but for $n = 2$ there is a simple procedure to find the signs of the eigenvalues by hand (using that the roots r, s of $x^2 + ux + v$ satisfy $r + s = -u$ and $rs = v$):

Theorem 26.3.3. Consider a symmetric 2×2 matrix $A = \begin{bmatrix} a & b \\ b & d \end{bmatrix}$, so the eigenvalues are the roots of the characteristic polynomial (from Theorem 23.3.1) $P_A(\lambda) = \lambda^2 - \mathrm{tr}(A)\lambda + \det(A)$, where (as *defined* in Theorem 23.3.1) $\mathrm{tr}(A) = a + d$ and $\det(A) = ad - b^2$. For its two roots $\lambda_1, \lambda_2 \in \mathbf{R}$:

(i) λ_1, λ_2 have opposite signs precisely when the product $\det(A) = \lambda_1\lambda_2$ is negative, so the indefinite case occurs exactly when $ad - b^2 = \det(A) < 0$;
(ii) λ_1, λ_2 are either both positive or both negative precisely when their product $\det(A) = \lambda_1\lambda_2$ is positive, so A is positive-definite or negative-definite precisely when $ad - b^2 = \det(A) > 0$;
(iii) in case (ii), the common sign of λ_1 and λ_2 is the same as that of their sum $\lambda_1 + \lambda_2 = \mathrm{tr}(A) = a + d$.

Example 26.3.4. Let's return to the examples in (24.2.1). The quadratic form $10x_1^2 - 14x_1x_2 + 5x_2^2$ is $q_A(x_1, x_2)$ for $A = \begin{bmatrix} 10 & -7 \\ -7 & 5 \end{bmatrix}$, with $\det(A) = 50 - 49 = 1 > 0$ and $\mathrm{tr}(A) = 15 > 0$, so A and q_A are positive-definite. The quadratic form $8x_1^2 - 10x_1x_2 + 3x_2^2$ is $q_B(x_1, x_2)$ for $B = \begin{bmatrix} 8 & -5 \\ -5 & 3 \end{bmatrix}$, with $\det(B) = 24 - 25 = -1 < 0$, so B and q_B are indefinite (e.g., $q_B(1,1) = 1 > 0$, $q_B(2,3) = -1 < 0$). ■

Remark 26.3.5. Now we can finally explain the observation in Example 10.2.13 that contour plots near critical points look like approximate nested tilted ellipses near local extrema (with the behavior of the numerical labels on the level curves as we approach the critical point – increasing or decreasing – telling us whether it is a local maximum or local minimum) and look like approximate tilted hyperbolas near saddle points (with the level curve through the saddle point approximately an "X" shape). Both cases arose in Figure 26.1.3, and will be seen in many plots in Section 26.4 (e.g., Figures 26.4.1 and 26.4.2).

For a critical point $\mathbf{a}$ of $f(x,y)$, we know that $(\mathrm{H}f)(\mathbf{a})$ admits an orthogonal basis of eigenvectors $\{\mathbf{w}_1,\mathbf{w}_2\}$. Let $\mathbf{w}_i' = \mathbf{w}_i/\|\mathbf{w}_i\|$, so $\{\mathbf{w}_1',\mathbf{w}_2'\}$ is an *orthonormal* basis of eigenvectors. Assume the corresponding eigenvalues λ_1,λ_2 are *nonzero* (this is the typical case, and avoids degenerate situations as in Figure 26.2.3). Combining the quadratic approximation for f near $\mathbf{a}$ with the diagonalization formula (24.2.2) applied to $q_{(\mathrm{H}f)(\mathbf{a})}(\mathbf{x})$, for h_1,h_2 near 0 we have

$$f(\mathbf{a}+h_1\mathbf{w}_1'+h_2\mathbf{w}_2') \approx f(\mathbf{a}) + \frac{\lambda_1(\mathbf{w}_1'\cdot\mathbf{w}_1')}{2}h_1^2 + \frac{\lambda_2(\mathbf{w}_2'\cdot\mathbf{w}_2')}{2}h_2^2 = f(\mathbf{a}) + \frac{\lambda_1}{2}h_1^2 + \frac{\lambda_2}{2}h_2^2.$$

Using the right side in place of f near $\mathbf{a}$, level curves $f = c$ are *well-approximated near* $\mathbf{a}$ by level curves $f(\mathbf{a}) + \frac{\lambda_1}{2}h_1^2 + \frac{\lambda_2}{2}h_2^2 = c$ for (h_1,h_2) near $(0,0)$ and c near $f(\mathbf{a})$. By subtracting $f(\mathbf{a})$ from both sides and multiplying by 2 throughout, we get an approximate description of level curves of f near $\mathbf{a}$:

$$\lambda_1 h_1^2 + \lambda_2 h_2^2 = 2(c - f(\mathbf{a})) \tag{26.3.1}$$

for h_1,h_2 near 0. This changes the *numerical label* on a level curve (c is replaced with $2(c-f(\mathbf{a}))$) but not the actual level curves.

[Note that $\|h\mathbf{w}_i\| = |h|\|\mathbf{w}_i\|$ whereas $\|h\mathbf{w}_i'\| = |h|$. Thus, to encode the precise geometry of the level curves for f near $\mathbf{a}$ (i.e., avoiding a distortion factor of $\|\mathbf{w}_i\|$ in distance) and not merely the qualitative aspects of being approximate ellipses or hyperbolas centered at $\mathbf{a}$ with specific lines of symmetry through $\mathbf{a}$, we use the *unit* eigenvectors $\mathbf{w}_i' = \mathbf{w}_i/\|\mathbf{w}_i\|$ for work with $(\mathrm{H}f)(\mathbf{a})$ as done above.]

The shape of (26.3.1) depends on the signs of the λ_i's, as we now discuss. For any $A, B \neq 0$ the level curves of $Ah_1^2 + Bh_2^2$ are ellipses centered at (0,0) when $A, B > 0$ or $A, B < 0$ and are hyperbolas centered at (0,0) when A and B have opposite signs, with the level curve through the origin in the latter case given by a pair of crossing lines (an "X" shape). We arrived at the left side of (26.3.1) by working with $f(\mathbf{a}+h_1\mathbf{w}_1'+h_2\mathbf{w}_2')$, so h_i keeps track of motion through $\mathbf{a}$ in the direction of $\mathbf{w}_i'$ (or equivalently of $\mathbf{w}_i$). Hence, the ellipses and hyperbolas in (26.3.1) approximating level curves of f near $\mathbf{a}$ are centered at $\mathbf{a}$ (corresponding to $(h_1,h_2) = (0,0)$) and are aligned with the *perpendicular eigenline directions* (for $(\mathrm{H}f)(\mathbf{a})$) through $\mathbf{a}$ (generally **not** the coordinate-axis directions through $\mathbf{a}$). These are illustrated in Figure 26.3.1 for two specific quadratic forms that were addressed in detail in Examples 25.4.2 and 25.4.3.

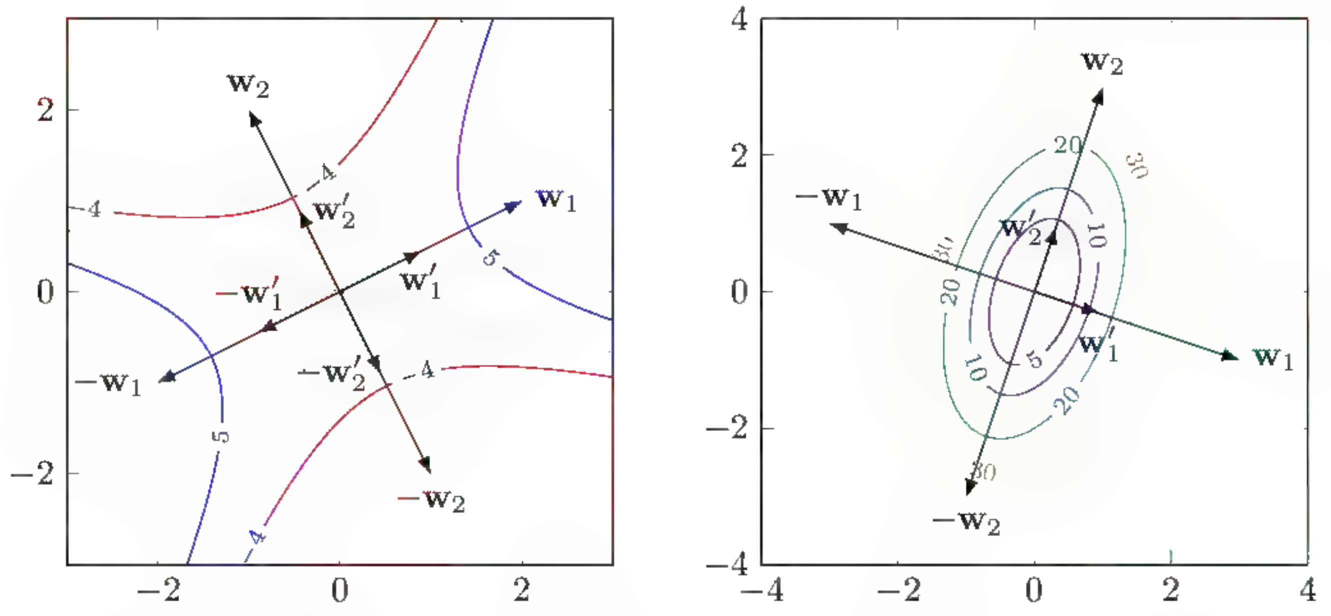


FIGURE 26.3.1. The level sets $x^2+4xy-2y^2 = 5$ (blue) and $x^2+4xy-2y^2 = -4$ (red) on the left are tilted versions of $2x^2-3y^2 = 5$ and $2x^2-3y^2 = -4$ respectively. The level curves of $13x^2-6xy+5y^2$ on the right are tilted versions of level curves of $14x^2+4y^2$.

Remark 26.3.6. There is a *third way* to formulate the Second Derivative Test when $n = 2$, given in most multivariable calculus books. Version II above is often the most useful formulation, since the eigenvalues usually contain valuable information (e.g., see Appendix J for applications in chemistry), so it is Version II that you are expected to know for homework and exams. The statement of a "Version III" for the 2-variable case can be given without any linear algebra (which is why other multivariable calculus books use it), but we don't give it since that would not promote any real understanding.

The formulation of Version III for $n = 2$ without linear algebra is cryptic, but its analogue for general n – called *Sylvester's Criterion* for (semi-)definiteness of symmetric $n \times n$ matrices – involves $k \times k$ determinants for all $k \leq n$. This criterion arises in economics (e.g., to ensure "positivity" of the solution to certain linear systems [KMS]), in thermodynamics (e.g., to analyze convexity of an energy function [Hor, pp. 200-201]), in other areas of math, etc. The proof of Sylvester's Criterion in Wikipedia is dreadful, but [Gilb] gives an elegant treatment in general (any n). It is impossible to sensibly explain where Version III for $n = 2$ comes from without linear algebra, and its equivalence with the $n = 2$ case of Version I in Theorem 26.1.5 (which is crucial) uses Version II above.

26.4. Using definiteness and eigenvalues to analyze critical points. We now use Theorem 26.3.3 to understand the behavior of some 2-variable functions near critical points, and also apply the eigenvalue method to a 3-variable function with non-diagonal Hessian at two critical points.

Example 26.4.1. Let's work out the behavior of $f(x, y) = x^3 - 3x^2 - 6xy + 9x + 3y^2$ at each of its critical points. The partial derivatives are

$$f_x = 3x^2 - 6x - 6y + 9, \quad f_y = -6x + 6y,$$

so the vanishing of f_y forces $x = y$, in which case the vanishing of f_x amounts to the quadratic condition $3x^2 - 12x + 9 = 0$, or in other words $3(x^2 - 4x + 3) = 0$. This factors as $3(x - 1)(x - 3) = 0$, so the critical points are $(1, 1)$ and $(3, 3)$, which are marked as black dots in both the contour plot and surface graph for $f(x, y)$ in Figure 26.4.1.

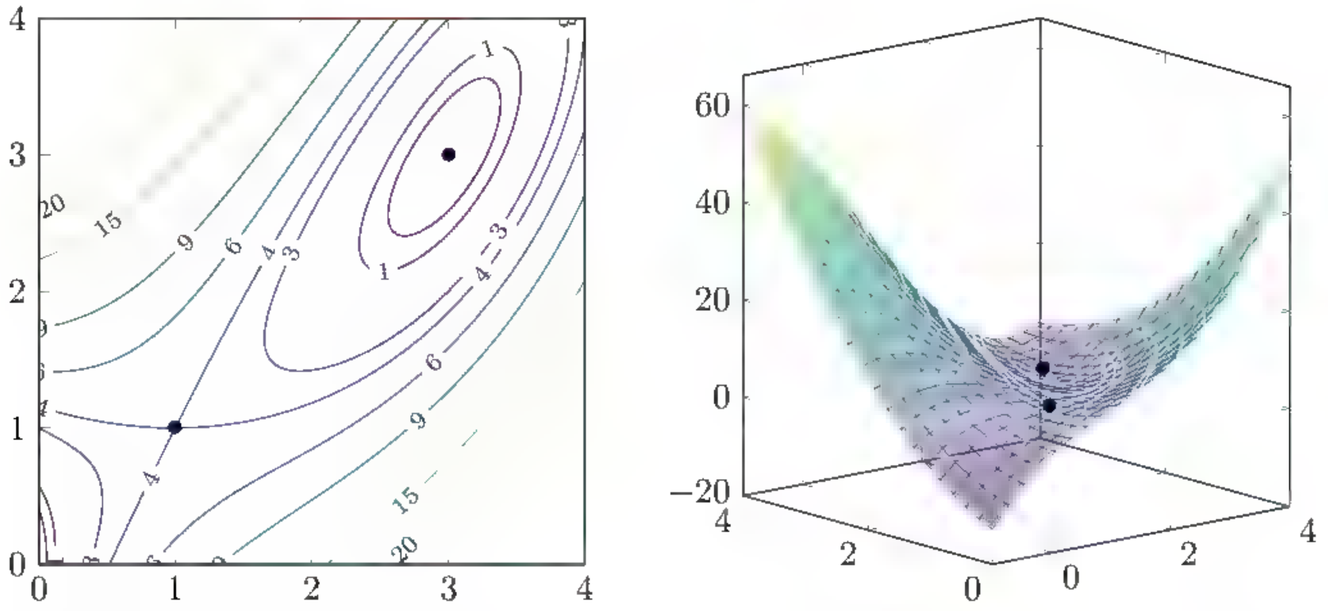


FIGURE 26.4.1. Contour plot and graph of $f(x, y) = x^3 - 3x^2 - 6xy + 9x + 3y^2$

The general Hessian for f is

$$(\mathrm{H}f)(x, y) = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{bmatrix} = \begin{bmatrix} 6x - 6 & -6 \\ -6 & 6 \end{bmatrix},$$

so

$$(\mathrm{H}f)(1,1) = \begin{bmatrix} 0 & -6 \\ -6 & 6 \end{bmatrix}, \quad (\mathrm{H}f)(3,3) = \begin{bmatrix} 12 & -6 \\ -6 & 6 \end{bmatrix}.$$

These are not diagonal, so we will use Theorem 26.3.3. We compute

$$\det(\mathrm{H}f)(1,1) = 0 - 36 = -36 < 0, \quad \det(\mathrm{H}f)(3,3) = 72 - 36 = 36 > 0,$$

so by Theorem 26.3.3(ii) the Hessian $(\mathrm{H}f)(1,1)$ is indefinite (making $(1,1)$ a saddle point of f) and the Hessian $(\mathrm{H}f)(3,3)$ is definite: either positive-definite or negative-definite.

To determine the common sign of the eigenvalues of $(\mathrm{H}f)(3,3)$ (and hence the type of definiteness), we compute its trace to be $12 + 6 = 18 > 0$, so by Theorem 26.3.3(iii) the eigenvalues are positive and hence this Hessian is positive-definite. This shows that $(3,3)$ is a local minimum for f. Observe that the geometry of the contour plot near both critical points exactly matches what we expect from Remark 26.3.5. (We could also compute the eigenvalues of each Hessian, but it isn't necessary here.) ■

Example 26.4.2. In Example 11.3.5 we ran gradient descent for the function

$$f(x,y) = x^2 - 3xy + 3y^2 + 5y + 2x,$$

seeking a local minimum. We eventually arrived at a (numerical approximation to a) critical point. We want to justify that this point is a local minimum (and not, say, a saddle point) by analyzing definiteness properties of the Hessian there.

In this case we get "lucky" insofar as the Hessian winds up being a constant matrix (i.e., the same at all points (x,y)), ultimately because f is a polynomial in x and y each of whose terms $x^i y^j$ has total degree (i.e., sum $i + j$ of exponents of x and y) at most 2. This implies that the definiteness properties of this are the same everywhere. To compute the Hessian, we need to compute the second partial derivatives of f. We begin by computing (as we already did in Example 11.3.5) the first partial derivatives, which is to say the gradient:

$$(\nabla f)(x,y) = \begin{bmatrix} 2x - 3y + 2 \\ -3x + 6y + 5 \end{bmatrix}.$$

Now differentiating these vector entries with respect to x and y yields

$$(\mathrm{H}f)(x,y) = \begin{bmatrix} 2 & -3 \\ -3 & 6 \end{bmatrix}$$

(independent of (x,y) for this f, as promised).

This Hessian has determinant $12 - 9 = 3 > 0$, so by Theorem 26.3.3(ii) it is definite: either positive-definite or negative-definite. To determine the common sign of the eigenvalues of this Hessian, we compute the trace to be $2+6 = 8 > 0$, so by Theorem 26.3.3(iii) the eigenvalues are positive and hence the Hessian is positive-definite. This shows that the (approximate) critical point we found in Example 11.3.5 is a local minimum, as desired. ■

> Always remember, as in Example 26.4.2, that **only at a critical point** does definiteness of the Hessian imply the point is a local extremum; elsewhere definiteness implies *nothing* about extrema (just like the single-variable second derivative test). The approximation (26.1.2) is *only* for critical points.

Example 26.4.3. As an application of Theorem 26.3.3(ii),(iii), we can finish off the analysis of definiteness in Example 26.2.5 (a setting that arises in economics). For $\lambda > 0$ we saw that the function $P(x,y) = x^{1/4}y^{1/2} - \lambda(x + y - 3/4)$ (with $x, y > 0$) has

$$(a,b) = (1/(64\lambda^4), 1/(32\lambda^4))$$

as its only critical point, and we worked out the Hessian:

$$(\mathrm{H}P)(x,y) = \begin{bmatrix} -\frac{3}{16}x^{\ 7/4}y^{1/2} & \frac{1}{8}x^{\ 3/4}y^{\ 1/2} \\ \frac{1}{8}x^{\ 3/4}y^{\ 1/2} & -\frac{1}{4}x^{1/4}y^{\ 3/2} \end{bmatrix}. \tag{26.4.1}$$

Our task is to figure out if $(\mathrm{H}P)(a,b)$ is positive-definite, negative-definite, indefinite, or "none of the above", with the aim of determining if (a,b) is a maximum or minimum (or something else). We do this in two ways: working directly at (a,b) and computing the eigenvalues of $(\mathrm{H}P)(a,b)$ explicitly, and working at a general (x,y) using just sign information for the determinant and trace of $(\mathrm{H}P)(x,y)$. In both approaches, we will see that $(\mathrm{H}P)(a,b)$ is negative-definite, so (a,b) is a local maximum for P.

Method I. For the explicit method working at (a,b), we compute the eigenvalues of $A = (\mathrm{H}P)(a,b)$. The characteristic polynomial is $t^2 - \mathrm{tr}(A)t + \det(A)$, and from the explicit description of A found in (26.2.1) we compute $\det(A) = 2(16\lambda^5)^2$ and $\mathrm{tr}(A) = -4(16\lambda^5)$. Hence, by the quadratic formula the eigenvalues are $16\lambda^5(-2 \pm \sqrt{2})$. These are both negative since $\lambda > 0$, so $(\mathrm{H}P)(a,b)$ is negative-definite.

Method II. For the alternative, less explicit, method that focuses just on the sign information of the determinant and trace of the Hessian, we will work at all (x,y) with $x,y > 0$. First, we compute the determinant of the Hessian in (26.4.1) at a general point (x,y). It might seem as if this will become a mess, but if you look at the diagonal and off-diagonal pairs in (26.4.1) then you'll see that the product of each such pair has the *same* exponent for each of x and y, so it won't be such a mess after all. More explicitly, the determinant is $(3/64)x^{\ 3/2}y^{\ 1} - (1/64)x^{\ 3/2}y^{\ 1} = (1/32)x^{\ 3/2}y^{\ 1}$. This is always positive on the region of interest, so we know that the eigenvalues are nonzero with the same sign; correspondingly the Hessian is either positive-definite or negative-definite.

To figure out if the common sign is positive or negative, in accordance with Theorem 26.3.3(iii) we compute the trace (sum of diagonal entries): this is negative since both diagonal entries of the Hessian are negative, so the eigenvalues are negative and hence the Hessian is negative-definite.

Here is a proof that (a,b) is a *global* maximum. First, $P(a,b) = 1/(64\lambda^3) + (3/4)\lambda > (3/4)\lambda$ and along the "boundary" (the non-negative coordinate axes), we have $P(x,0) = -\lambda x + (3/4)\lambda < (3/4)\lambda < P(a,b)$ and $P(0,y) = -\lambda y + (3/4)\lambda < (3/4)\lambda < P(a,b)$. Finally, on each line $y = cx$ ($c > 0$), we have $P(x,cx) = \sqrt{c}x^{3/4} - \lambda(1+c)x + (3/4)\lambda$ is $\le (3/4)\lambda < P(a,b)$ whenever $x \ge c^2/(\lambda(1+c))^4$, so if $P(x,cx) \ge P(a,b)$ then $x < c^2/(\lambda(1+c))^4 < 1/\lambda^4$ (so $y = cx = c^3/(\lambda(1+c))^4 < 1/\lambda^4$). Hence, any (x_0,y_0) with $P(x_0,y_0) \ge P(a,b)$ is inside the curve of points $(c^2/(\lambda(1+c))^4, c^3/(\lambda(1+c))^4)$ in the interior of the rectangle $R = \{(x,y) : 0 \le x,y \le 1/\lambda^4\}$. A continuous function on a rectangle attains a maximum (an $\mathbf{R}^2$-analogue of the Extreme Value Theorem), so P attains a global maximum in the interior of R. This is a critical point, so it is (a,b).

■

Example 26.4.4. We now treat a 3-variable function. This is *absolutely impossible* to analyze via the 2-variable methods of staring at contour plots or surface graphs, so it illustrates the full glory of eigenvalue analysis of Hessians.

For $f(x,y,z) = xy + yz + xz + xyz$, we seek the critical points of f and the behavior of f near each. We compute the first partial derivatives to be

$$f_x = y + z + yz, \quad f_y = x + z + xz, \quad f_z = x + y + xy,$$

so the vanishing of these can be expressed as the combined conditions

$$y(1+z) = -z, \quad z(1+x) = -x, \quad y(1+x) = -x. \tag{26.4.2}$$

We can't have $1 + x = 0$ since this says $x = -1$ yet either the second or third equations in (26.4.2) would then say $0 = 1$, an absurdity. Thus, we can divide by $1 + x$ to get $z = -x/(1+x) = y$. Plugging $z = y$

into the first equation gives $y(1+y) = -y$, so either $y = 0$ or we can cancel y to obtain $1 + y = -1$, which is to say $y = -2$.

If $y = 0$ then also $z = y = 0$, and $x = 0$ by the third equation in (26.4.2). In this case we obtain $(0,0,0)$, at which all three partials vanish, so this is a critical point. If instead $y = -2$ then also $z = y = -2$, so the second or third equation in (26.4.2) gives $-2(1 + x) = -x$, which has as its only solution $x = -2$. So we arrive at the only other option $(-2,-2,-2)$ that is indeed checked to be a critical point.

From the first partials of f we compute the second partials, filling in the entries of the general Hessian:

$$(\mathrm{H}f)(x,y,z) = \begin{bmatrix} 0 & 1+z & 1+y \\ 1+z & 0 & 1+x \\ 1+y & 1+x & 0 \end{bmatrix}.$$

At the two critical points we obtain the Hessian matrices

$$(\mathrm{H}f)(0,0,0) = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad (\mathrm{H}f)(-2,-2,-2) = \begin{bmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix}.$$

The first of these was encountered near the start of Section 24.3, where we saw an orthogonal basis of eigenvectors $\mathbf{v}, \mathbf{v}', \mathbf{v}''$ whose respective eigenvalues are $-1, -1, 2$. These eigenvalues have mixed signs, so this Hessian is indefinite (even though its entries are ≥ 0!), so the critical point is a saddle point.

At the other critical point $(-2,-2,-2)$ the Hessian is the negative of the one obtained at $(0,0,0)$, so it has the *same* eigenvectors but with eigenvalues $1, 1, -2$ that have been multiplied by -1. This is still mixed signs, so again the critical point is a saddle point. Hence, f has no local extrema! ■

Example 26.4.5. We finish our tour of examples by revisiting our old friend $2x^2 - 5xy + 2y^3 + 3x^2y + 2y^2$ from Example 25.3.4, which had two nearby critical points: $P = (0,0)$ and Q. In the middle picture in Figure 26.4.2 below we provide a contour plot on the scale of the same square as seen in Example 25.3.4, namely the square of points (x, y) with $|x|, |y| \leq 2$. A glance at that plot suggests that something a bit peculiar may be happening near the origin, but it is hard to see.

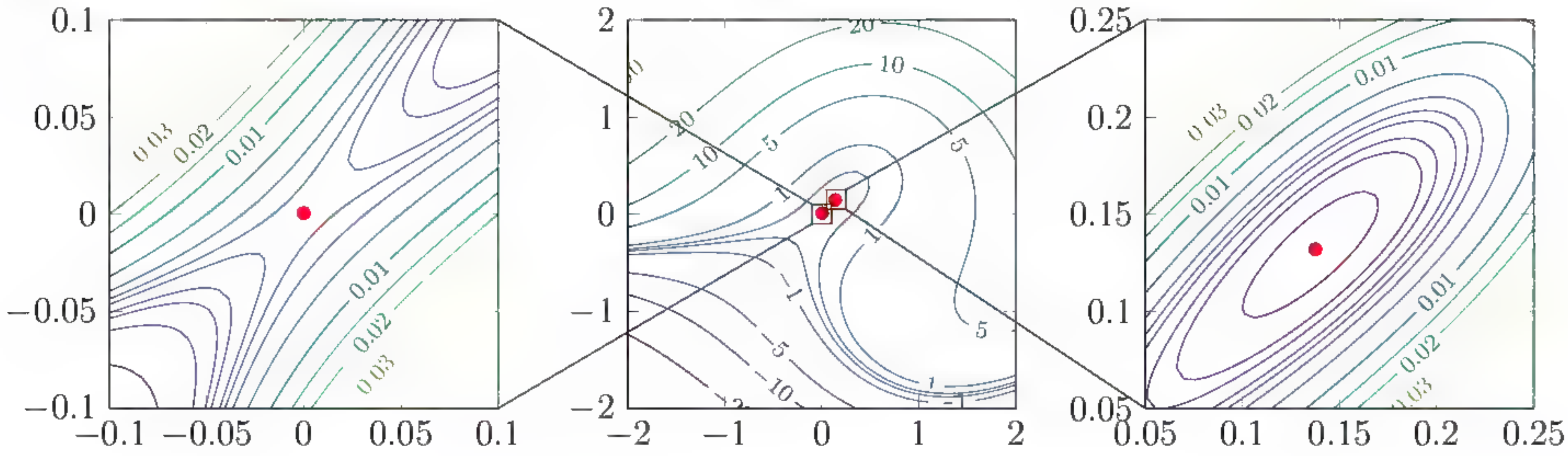


FIGURE 26.4.2. Contour plot of $f(x,y) = 2x^2 - 5xy + 2y^3 + 3x^2y + 2y^2$ with two rather close critical points, one a local minimum and one a saddle point, along with 25-fold zoom at each critical point to reveal the familiar type of contour plot (very!) nearby.

On the left and right sides in Figure 26.4.2, we see a 25-fold zoom on $P = (0,0)$ (red dot for the left picture) and the nearby critical point Q (red dot for the right picture) that reveals what is going on: part of the elliptic level sets near Q (as expected from Remark 26.3.5) is also part of the hyperbolic level sets near P! That explains the unusual-looking shape of the contour plot in the surrounding vicinity.

Let's now apply our systematic eigenvalue techniques to this function $f(x,y)$, to see that it detects the microscopic local behavior illustrated so strikingly in Figure 26.4.2. The gradient of f is

$$\nabla f = \begin{bmatrix} 4x-(5-6x)y \\ 3x^2-5x+(6y^2+4y) \end{bmatrix},$$

so any critical point (x,y) must satisfy the two equations

$$4x=(5-6x)y, \quad 3x^2-5x+(6y^2+4y)=0.$$

The first equation shows that if $x=0$ then $y=0$, and clearly $P=(0,0)$ satisfies both equations, so P is a critical point. For $x \neq 0$, the first equation forces $5-6x \neq 0$ and $y=4x/(5-6x)$. Plugging this into the second equation and doing some algebra (and cancelling a factor of $3x \neq 0$ that appears throughout) yields the cubic condition on x mentioned in Example 25.3.4. That cubic has only one real root a (which can be approximated on a computer); this gives the other critical point $Q=(a,b)$ with $b=4a/(5-6a)$.

To understand the local behavior of f near these points, we differentiate the components of ∇f to obtain the Hessian $(\mathrm{H}f)(x,y)=\begin{bmatrix} 4+6y & -5+6x \\ -5+6x & 4+12y \end{bmatrix}$. Thus, $(\mathrm{H}f)(0,0)=\begin{bmatrix} 4 & -5 \\ -5 & 4 \end{bmatrix}$ has determinant $16-25=-9<0$, so its eigenvalues have opposite signs and hence the Hessian is indefinite. This shows (0,0) is a saddle point, with level sets that are approximately hyperbolas *very nearby* (as seen in the 25-fold zoom on the left in Figure 26.4.2). For $Q=(a,b)$ we have

$$(\mathrm{H}f)(Q)=\begin{bmatrix} 4+6b & -5+6a \\ -5+6a & 4+12b \end{bmatrix},$$

and $\det(\mathrm{H}f)(Q)$ is approximated using numerical approximations to a and $b=4a/(5-6a)$. This determinant is positive, so the eigenvalues are nonzero with the same sign and $(\mathrm{H}f)(Q)$ is definite. Thus, the level sets *very nearby* Q are approximated by ellipses, as seen in the 25-fold zoom on the right in Figure 26.4.2. The trace is $8+18b>0$, so the eigenvalues are both positive and hence Q is a local minimum. ■

Remark 26.4.6. In this chapter we have seen the close relationship between the behavior of a function $f:\mathbf{R}^n \to \mathbf{R}$ near a critical point $\mathbf{a}$ and the eigenvalues and eigenvectors of the Hessian $(\mathrm{H}f)(\mathbf{a})$, especially when *all* eigenvalues are nonzero. Using the same mathematical input, it turns out that the nearly opposite extreme of a Hessian having only one nonzero eigenvalue which moreover has *large* absolute value with $n=2,3$ underlies the computer vision technique called ridge detection. This is broadly useful in areas such as: facial recognition with wrinkles, detecting blood vessels in brain imaging, and digital humanities to reconstruct (via scanning techniques) a virtual unfolded version of ancient documents that are too fragile to open up!

26.5. Your mathematical journey, and future course advice. After all of the hard work you put in during this course, you are now aware of a vast array of mathematical concepts and techniques that you didn't know before (some of which may require further practice and experience to fully internalize). Even if you have previously seen some concepts in this course (possibly in special cases), we hope that now you more deeply understand them and their relation to broader contexts. And you have seen that solving a mathematical problem can require the development of *new ways of thinking*, not only plugging something into a new formula. For example:

(i) When you hear a physicist talk about 4-dimensional geometry, a computer scientist speak about a million-dimensional data structure, or an economist refer to high-dimensional statistics, you now know that there is absolutely nothing mysterious or "purely theoretical" going on. The real world abounds in problems involving n unknowns with large n, and you are now familiar with the algebraic and geometric framework for seriously discussing such problems and even solving them.
Humans invented both linear algebra in $\mathbf{R}^n$ (for all n) and associated concepts such as distance, angle, subspace, and especially *dimension* that provide a powerful visual language to discuss a vast array

of real-world questions involving many unknowns. This language works so well that it enables us to develop visual expectations (which can often be proved as theorems) far beyond the cases $n = 2, 3$ of everyday experience.

(ii) You are now aware that solutions to practical mathematical problems can rely upon a surprising *synthesis* of concepts that were invented for other purposes. For instance, the story of the multivariable second derivative test for optimization of scalar-valued functions involved many algebraic and geometric ideas (quadratic forms, orthogonality, equality of mixed partials, and eigenvalues) which were each useful in their own way and seemed to have nothing to do with each other at first sight. Those ideas unexpectedly came together via a deep mathematical result (the Spectral Theorem).

If you look back over the table of contents, we hope you're pleased at how much you've grown mathematically in one quarter in terms of both knowledge and maturity of mathematical thinking. Learning is a process of successive approximation, and as you see the ideas and methods of this course arise in other areas of study you will solidify and deepen your command of them. In the final (optional) Chapter 27, we provide a taste of a few of the many additional ways in which the linear algebra ideas you have learned can be put to good use.

This course provides the background for many other Math classes that are useful in future studies either within mathematics or in other fields (e.g., natural sciences, engineering, economics, computer science, etc.). We'd like to offer some advice on choosing between two courses that are natural follow-ups: Math 104 and Math 113 (both offered every autumn, winter, and spring), each of which develops linear algebra in new but quite different directions.

Both Math 104 and Math 113 will give you a broader understanding of how to work with matrices and much more about the fundamental concepts of eigenvalues and eigenvectors than there is time to cover in Math 51 (e.g., generalized eigenspaces and Jordan form). The role of eigenvalues is so central in so many applications of linear algebra in the natural sciences, economics, statistics, and computer science that the utility of further study of linear algebra is likely to be closely linked to the extent to which it broadens your knowledge about eigenvalues (e.g., the workhorse of data science called "principal component analysis" is ultimately a statistical reinterpretation of the Spectral Theorem).

Math 104: One theme in Math 51 is that modern techniques for analyzing data, no matter the discipline in which they arise, rely on large-scale linear algebra computations. You now know that understanding the basic principles of linear algebra in $\mathbf{R}^2$ or $\mathbf{R}^3$ is not so different from $\mathbf{R}^n$ for possibly very large values of n when one has set up an appropriate language.

However, when it comes to doing actual computations, there is a big difference between smaller and larger dimensions. When doing calculations in $\mathbf{R}^n$, say when $n \leq 5$, it is not particularly laborious for a computer to solve linear systems by the process of eliminating variables (called *Gaussian elimination*, expressed in more modern terms as an LU-decomposition), or to compute determinants to check whether matrices are invertible, etc. These techniques become less and less feasible on a computer when n is large. In real-world problems, n can be on the scale of millions or billions, and then computer calculations must be done with great forethought.

There is an entirely different side to applied linear algebra: finding efficient and quick (and numerically stable) algorithms for work in very large dimensions. Math 51 has sometimes mentioned the importance of numerical stability (e.g., when talking about the QR-decomposition), but there is a lot more to the story. The emphasis in Math 104 is on acquiring practical and conceptual fluency with some of the most important techniques and algorithms in applied linear algebra.

For example, in practice it is sometimes important to be able to estimate "how long" a given computation takes. This is the beginning of the study of computational complexity. If you are solving a problem involving an $n \times n$ matrix, then does the computation need only around n steps (this is regarded as very good), or n^3 steps (less good, but still quite tolerable), or 10^n steps (an utter disaster)?

Also, many problems simply cannot be handled "exactly". For example, the computation of the eigenvalues of an $n \times n$ matrix involves finding the roots of a specific polynomial of degree n (the "characteristic polynomial"), and not only is there no exact formula for those roots (for $n \geq 5$), but once n is even of moderate size it is not numerically feasible to use that polynomial to find the eigenvalues. Instead, one employs efficient and clever algorithms for numerically approximating the eigenvalues (as sketched in Appendix H, using matrix factorizations): a sequence of matrices is devised that allows one to create sequences of numbers converging to the eigenvalues of the original matrix.

Such material and much more (such as a broader mastery of eigenvalues and the related fundamental "singular value decomposition" introduced in Section 27.3) is covered in Math 104 and is essential for applications of linear algebra throughout data science, natural sciences, and engineering. There is a lot of excellent software to implement such algorithms, but the best scientists and engineers have a good sense of what is going on inside these software tools because in any real-life situation it is only a matter of time before one needs to make a computation or solve a problem for which the software tools in hand are not good enough and one needs to dig deeper to make things work. This is one among many reasons for learning the computational theory of linear algebra with the breadth and depth developed in Math 104.

Math 113: Further coursework in pure as well as many parts of applied mathematics involves knowing how to read and write proofs. Math 51 has given you a flavor of how the conceptual side of math (such as the importance of precise definitions and the utility of thinking in terms of properties rather than explicit numbers) can involve ways of thinking that are rather different from how you probably encountered math before now.

Gaining familiarity and facility with proof-writing and reading is best done in the context of using those skills to understand a substantive subject (just as learning how to cook is best done by preparing actual meals). There are several classes that the Math department offers which have proof-writing as one of their course-level learning goals: Math 110 (proofs in the context of applied number theory and cryptography, offered once each year), Math 115 (proofs of the core theorems of single-variable calculus, offered every autumn and spring), and Math 113 (proofs in the context of linear algebra, taught in every quarter). In Math 113 you will also deepen your knowledge about linear algebra, including a lot more about the "conceptual perspective" (building on notions and results that are already familiar to you from Math 51), and see how to push these ideas in many new directions.

Math 113 provides an introduction to proof-writing in the context of linear algebra developed in a manner that focuses on the logical structure of the subject, using the broader setting of vector spaces (with complete proofs throughout) and going much further. The types of reasoning involved in doing this are at the heart of nearly all contemporary mathematics, and are tremendously useful in parts of other disciplines too (e.g, theoretical computer science, quantum computation, quantitative finance, theoretical physics, computer graphics). For example, in applications of linear algebra throughout other quantitative fields as well as in computer science and in more advanced mathematics it is useful to work with scalars from number systems other than $\mathbf{R}$ (e.g., complex numbers, or "finite fields"), and the conceptual approach in Math 113 provides linear algebra in this broader setting. Another context where the conceptual perspective is important is investigating the question "What are the theoretical limits of anonymity when I am making queries on a large data set"? For example, can one get useful data out of large collections of genomic information, or even just health records, without compromising the anonymity of the people from whom this data was collected? (This is a field called "differential privacy".)

Everyone on the Math 51 course staff is happy to talk with you in person if you'd like to discuss these or other course options in more detail.

Chapter 26 highlights (link to highlights in previous chapter)

Notation	**Meaning**	**Location in text**
nothing new!		

Concept	**Meaning**	**Location in text**
saddle point for $f : \mathbf{R}^n \to \mathbf{R}$, with $n \geq 2$	a **critical point** $\mathbf{a}$ at which f has a local maximum on one line through $\mathbf{a}$ and a local minimum on another line through $\mathbf{a}$	Definition 26.1.3

Result	**Meaning**	**Location in text**
second derivative test via definiteness	a **critical point** $\mathbf{a}$ of f is a local maximum, local minimum, or saddle point if $(\mathrm{H}f)(\mathbf{a})$ is negative-definite, positive-definite, or indefinite respectively	Theorem 26.1.5
second derivative test via eigenvalues	a **critical point** $\mathbf{a}$ of f is a local maximum, local minimum, or saddle point if the eigenvalues of $(\mathrm{H}f)(\mathbf{a})$ are all negative, all positive, or at least one positive and one negative respectively	Theorem 26.3.1
for $n = 2$, determine signs of eigenvalues without computing the actual eigenvalues	sign of determinant indicates when eigenvalues have same or opposite signs, and when same sign then that sign is the same as the sign of the trace	Theorem 26.3.3
for $n = 2$, eigenvector information for Hessian controls contour plot near **critical point** $\mathbf{a}$	if eigenvalues have same sign then plot near $\mathbf{a}$ is approximate nested ellipses aligned with the eigenvector directions; if eigenvalues have opposite signs then plot near $\mathbf{a}$ is approximate nested hyperbolas aligned with the eigenvector directions	Remark 26.3.5

Skill	**Location in text**
for $n = 2$, be aware of how geometry of contour plot near a **critical point** indicates if it is a local extremum or a saddle point	Figures 26.1.3, 26.1.4, 26.3.1, 26.4.1
apply second derivative test via definiteness for *diagonal* Hessian at **critical points**	Example 26.2.4
recognize definiteness of diagonal matrices by inspection	Example 26.1.8
apply second derivative test via eigenvalues, especially for $n = 2$ (and when given orthogonal basis of eigenvectors or their eigenvalues for $n > 2$)	Examples 26.4.1, 26.4.2, 26.4.3, 26.4.1
for $n = 2$, use eigenvectors and eigenvalues for Hessian at **critical point** to sketch qualitatively correct contour plot nearby (nested ellipses or hyperbolas aligned with eigenvector lines)	(26.3.1), Figure 26.3.1

27. More eigenvalue applications: ODE systems, population dynamics, SVD (optional)

In this final (**optional!**) chapter we discuss more applications of eigenvalues and eigenvectors. We begin in Section 27.1 by using eigenvalues to transform linked systems of differential equations (as arise in the study of structures consisting of many moving parts in physics, economics, etc.). This leads to a systematic method for explicitly solving such differential equations, as you will explore in Math 53.

Next, returning in Section 27.2 to the topic of Markov chains that we introduced in Chapter 16, we explore more deeply how eigenvalues illuminate the behavior of matrix powers for large exponents that we initially considered in Section 24.4 for population dynamics. The key lesson is that the eigenvalues that are largest and second largest (in the sense of absolute value) control the long-term dynamics.

Finally, as an introduction to central concepts in machine learning and applications of data analysis throughout many scientific fields, in Section 27.3 we discuss the *singular value decomposition* of general matrices (possibly not symmetric, nor even square). This is a computationally powerful replacement for the notion of eigenvalues, though its development ultimately rests on a solid understanding of eigenvalues (and particularly the Spectral Theorem from Section 24.1). We discuss a variety of applications of the singular value decomposition, including *principal component analysis* (whose underlying mathematical structure amounts to a statistical perspective on the singular value decomposition).

By the end of this chapter, you should be able to:

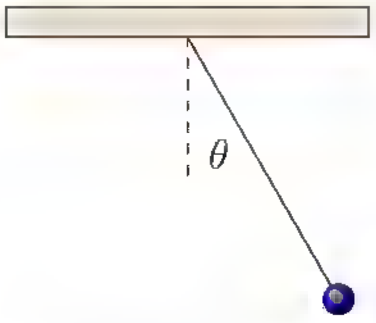
- use eigenvalues and eigenvectors to transform a pair of coupled (linear) differential equations into a pair of decoupled ones, at least when the governing 2×2 matrix is symmetric;
- use the largest eigenvalue and an associated eigenvector to approximate the long-term behavior of a discrete dynamical system, at least when the governing matrix is symmetric;
- define the singular value decomposition of a matrix, and relate this to rank-reduction and principal component analysis.

27.1. Coupled oscillators. In physics one often studies oscillating systems. For example, the oscillation of a pendulum whose initial angle is not too big can be (approximately) described by the differential equation

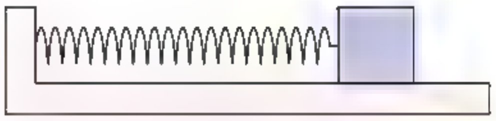
$$\frac{d^2\theta}{dt^2} + \frac{g}{\ell}\theta = 0, \tag{27.1.1}$$

where $g \approx 9.8\,\mathrm{ms}^{-2}$ is the gravitational acceleration, ℓ is the length of a pendulum, and θ is the angle that the pendulum makes with the vertical direction.

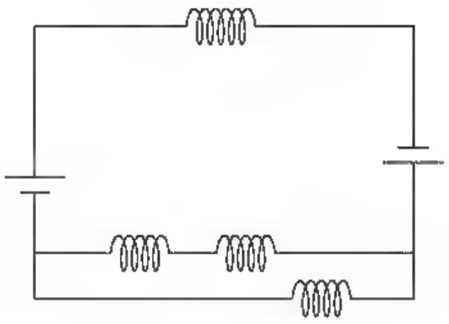
Other examples of oscillators include a mass on a spring (for which the role of θ is played by a variable representing the position of the mass) and electrical current through a circuit that contains capacitors and inductors (for which the role of θ is played by the current through the circuit).



System A: Simple pendulum.

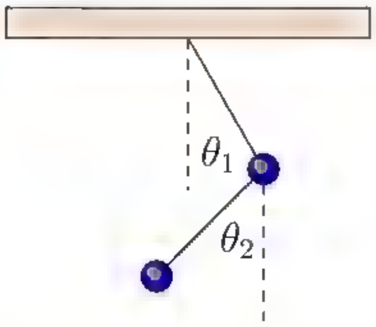


System B: Mass and spring.



System C: An electrical circuit.

When two such systems are coupled together – a simple example is the *double pendulum* – the result is still described by a differential equation, but now involving multiple variables.



System D: Double pendulum.

Namely, if θ_1, θ_2 are the angles describing the double pendulum, they will satisfy (approximately) a system of differential equations linking θ_1 and θ_2 that may look something like this:

$$\begin{cases} \dfrac{d^2\theta_1}{dt^2} + 3\theta_1 - 2\theta_2 = 0 \\ \\ \dfrac{d^2\theta_2}{dt^2} - 2\theta_1 + 3\theta_2 = 0. \end{cases} \tag{27.1.2}$$

This expresses how the second derivative of each θ_j depends on *both* θ_1 and θ_2; it is a *linked* system of differential equations. The specific coefficients in (27.1.2) depend on the ratios of the lengths and of the masses of the pendulums (though their signs match what is in (27.1.2)). We are not going to solve this system, but rather will explain how to *reduce* it to a pair of *unlinked* simpler equations like (27.1.1).

First of all, we are going to solve this directly, using a magical substitution. Then we will see that the substitution really came from eigenvectors (and so can be found in a much wider range of settings without any recourse to magic). As a magical substitution, consider the new variables u_1, u_2 given by

$$u_1 = \theta_1 + \theta_2, \quad u_2 = \theta_2 - \theta_1.$$

The values u_1, u_2 determine the angles θ_1, θ_2 since $\theta_1 = (u_1 - u_2)/2$ and $\theta_2 = (u_1 + u_2)/2$, so these new variables are just another way of encoding the angles of the pendulum (but they lack the direct physical significance of θ_1, θ_2). However, *mathematically* they will be very convenient.

To reveal the useful feature of these new variables, we compute their second derivatives:

$$\frac{d^2u_1}{dt^2} = \frac{d^2\theta_1}{dt^2} + \frac{d^2\theta_2}{dt^2} = -(3\theta_1 - 2\theta_2) - (-2\theta_1 + 3\theta_2) = -(\theta_1 + \theta_2) = -u_1, \tag{27.1.3}$$

$$\frac{d^2u_2}{dt^2} = \frac{d^2\theta_2}{dt^2} - \frac{d^2\theta_1}{dt^2} = -(-2\theta_1 + 3\theta_2) - (-3\theta_1 + 2\theta_2) = -5(\theta_2 - \theta_1) = -5u_2. \tag{27.1.4}$$

In other words, u_1, u_2 satisfy the equations $u_1'' + u_1 = 0, u_2'' + 5u_2 = 0$. *These equations are much simpler because u_1 and u_2 do not interact. They each look like the equation for a single pendulum.* In other words, by making a clever substitution, we were able (from the mathematical point of view) to "decouple" the two oscillators. The utility of such decoupling is that one can find general solutions u_1 and u_2 to each of these decoupled equations (this is studied from a wider perspective in Math 53), so then we can determine the general solutions for $\theta_1 = (u_1 - u_2)/2$ and $\theta_2 = (u_1 + u_2)/2$. (Up to a scaling factor, the same clever substitution arises in the quantum mechanical study of ammonia [**Feyn1**, Vol. III, Sec. 8.6, (8.46)-(8.53)].)

Now let us explain where this change of variables comes from, so we can see that it underlies a completely *systematic method* (not requiring clever tricks). Since the key theme of this book is always to use vectors and matrices to encode (systems of) equations, we first observe that the pair of differential equations (27.1.2) is the same as the single vector equation

$$\frac{d^2}{dt^2}\begin{bmatrix}\theta_1(t)\\ \theta_2(t)\end{bmatrix} + \underbrace{\begin{bmatrix}3 & -2\\ -2 & 3\end{bmatrix}}_{M}\begin{bmatrix}\theta_1(t)\\ \theta_2(t)\end{bmatrix} = \begin{bmatrix}0\\ 0\end{bmatrix}, \tag{27.1.5}$$

where we have used a shorthand: we write $\frac{d^2}{dt^2}\begin{bmatrix}\theta_1(t)\\ \theta_2(t)\end{bmatrix}$ to mean $\begin{bmatrix}\theta_1''(t)\\ \theta_2''(t)\end{bmatrix}$ (i.e., we differentiate a vector of functions entrywise). The key is to apply some serious linear algebra to the 2×2 matrix M of coefficients.

The matrix $M = \begin{bmatrix}3 & -2\\ -2 & 3\end{bmatrix}$ is symmetric (this happens for the double pendulum in special circumstances) with eigenvalues $\lambda_1 = 1$ and $\lambda_2 = 5$. Corresponding unit eigenvectors are

$$\mathbf{w}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 1\end{bmatrix},\quad \mathbf{w}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix}-1\\ 1\end{bmatrix}$$

(i.e., $M\mathbf{w}_1 = \mathbf{w}_1$ and $M\mathbf{w}_2 = 5\mathbf{w}_2$, as can be checked directly); the factor $1/\sqrt{2}$ in each $\mathbf{w}_j$ is there solely to make $\mathbf{w}_j$ a unit vector (i.e., $\|\mathbf{w}_j\| = 1$). As we have explained in Section 24.4, since the eigenvectors $\mathbf{w}_j$ are unit vectors, we have a matrix decomposition

$$M = \underbrace{\frac{1}{\sqrt{2}}\begin{bmatrix}1 & -1\\ 1 & 1\end{bmatrix}}_{W}\underbrace{\begin{bmatrix}1 & 0\\ 0 & 5\end{bmatrix}}_{D}\underbrace{\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ -1 & 1\end{bmatrix}}_{W^{-1}=W^{\top}}$$

where W is the matrix whose jth column is $\mathbf{w}_j$.

Remark 27.1.1. Note that W is an orthogonal matrix (i.e., $W^{\top}W = \mathrm{I}_2$) because the $\mathbf{w}_j$ are pairwise orthogonal unit vectors (as can be checked by hand). This is not a coincidence, but rather holds by design: if eigenvectors $\mathbf{v}, \mathbf{v}'$ of a symmetric matrix M have *different* eigenvalues then $\mathbf{v}$ and $\mathbf{v}'$ must be orthogonal (Theorem 24.1.1(ii)).

The vector differential equation (27.1.5) becomes:

$$\frac{d^2}{dt^2}\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} + WDW^{-1}\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} = \begin{bmatrix}0\\ 0\end{bmatrix}.$$

Now if we multiply on the left by the "constant matrix" W^{-1} (this has entries that are fixed numbers, having nothing to do with t) then it passes straight through the differentiation (by virtue of being a "constant matrix") to yield:

$$\frac{d^2}{dt^2}\left(W^{-1}\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}\right) + D\left(W^{-1}\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}\right) = \begin{bmatrix}0\\ 0\end{bmatrix}. \tag{27.1.6}$$

But what is $W^{-1}\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}$? Writing it out, we have

$$W^{-1}\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ -1 & 1\end{bmatrix}\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}\theta_1 + \theta_2\\ \theta_2 - \theta_1\end{bmatrix}.$$

Therefore – except for the scalar factor $1/\sqrt{2}$ – the matrix-vector product $W^{-1}\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}$ precisely recovers the variables u_1 and u_2 that we used above. Hence, upon multiplying through by $\sqrt{2}$ (to get rid of the factor $1/\sqrt{2}$ everywhere), (27.1.6) says exactly:

$$\frac{d^2}{dt^2}\begin{bmatrix}u_1\\ u_2\end{bmatrix} + \begin{bmatrix}1 & 0\\ 0 & 5\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix} = \begin{bmatrix}0\\ 0\end{bmatrix}.$$

Unraveling the matrix notation, this is *precisely* the simplified system (27.1.3) and (27.1.4) that we found earlier! The fact that D is *diagonal* corresponds to the decoupling with the non-interaction of u_1 and u_2. So the viewpoint of eigenvectors and eigenvalues indeed explains the original magical substitution u_1, u_2 in place of θ_1, θ_2 (apart from the scalar factor $1/\sqrt{2}$ which has *no impact* on the differential equation satisfied by each u_j).

The advantage of the approach via eigenvalues is that it is *systematic* and moreover works perfectly well even if we had 50 variables $\theta_1, \dots, \theta_{50}$ instead of 2 variables, provided of course that one knows how to find eigenvalues of a 50×50 matrix (which can be done on a computer with a high degree of accuracy in many circumstances). The moral remains the same:

> Using eigenvalues and eigenvectors, systems of many linked oscillators can be turned into a collection of independent oscillators (at least mathematically). The same happens for many other situations with lots of moving parts that influence each other as they evolve in time.

Example 27.1.2. To give another worked example, let's use the same technique to reduce the pair of linked differential equations

$$\frac{d^2x}{dt^2} = 2x + 6y, \quad \frac{d^2y}{dt^2} = 6x + 7y \tag{27.1.7}$$

to a pair of differential equations that are decoupled (i.e., each can be analyzed and solved separately). The method is sketched here, and we leave it to the reader to fill in the details of the omitted computational steps that proceed exactly as what we did for the system (27.1.2).

The system (27.1.7) can be written in vector form as

$$\frac{d^2}{dt^2}\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + \begin{bmatrix} -2 & -6 \\ -6 & -7 \end{bmatrix}\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

and the (symmetric) matrix

$$M = \begin{bmatrix} -2 & -6 \\ -6 & -7 \end{bmatrix}$$

that appears has as its eigenvalues $\lambda_1 = -11$ and $\lambda_2 = 2$. These have respective unit eigenvectors

$$\mathbf{w}_1 = \frac{1}{\sqrt{13}}\begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad \mathbf{w}_2 = \frac{1}{\sqrt{13}}\begin{bmatrix} -3 \\ 2 \end{bmatrix}$$

which are readily checked to be orthogonal to each other (as we know they must be since they correspond to *different* eigenvalues of a symmetric matrix); these are unit vectors due to the factor $1/\sqrt{13}$ in each.

The 2×2 matrix W with jth column $\mathbf{w}_j$ is therefore orthogonal; explicitly it is

$$W = \frac{1}{\sqrt{13}}\begin{bmatrix} 2 & -3 \\ 3 & 2 \end{bmatrix},$$

and $W^{-1} = W^\top$ by orthogonality. Hence, the magical change of variables which will decouple the pair of differential equations is given by

$$W^{-1}\begin{bmatrix} x \\ y \end{bmatrix} = W^\top \begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{\sqrt{13}}\begin{bmatrix} 2 & 3 \\ -3 & 2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{\sqrt{13}}\begin{bmatrix} 2x + 3y \\ -3x + 2y \end{bmatrix}.$$

As in our analysis of (27.1.2), the overall scaling factor $1/\sqrt{13}$ is irrelevant for the purpose of decoupling the differential equations, so we are led to introduce the new functions

$$u_1(t) = 2x(t) + 3y(t), \quad u_2(t) = -3x(t) + 2y(t)$$

(from which we can recover the original functions: $x = (2u_1 - 3u_2)/13$ and $y = (3u_1 + 2u_2)/13$).

Sure enough, when we compute second derivatives we get the desired decoupling:

$$\frac{d^2u_1}{dt^2} = 2\frac{d^2x}{dt^2} + 3\frac{d^2y}{dt^2} = 2(2x+6y) + 3(6x+7y) = 22x + 33y = 11(2x+3y) = 11u_1,$$

$$\frac{d^2u_2}{dt^2} = -3\frac{d^2x}{dt^2} + 2\frac{d^2y}{dt^2} = -3(2x+6y) + 2(6x+7y) = 6x - 4y = -2(-3x+2y) = -2u_2.$$

In other words, the original linked pair of differential equations for x and y is equivalent to the pair of unlinked differential equations

$$\frac{d^2u_1}{dt^2} - 11u_1 = 0, \quad \frac{d^2u_2}{dt^2} + 2u_2 = 0$$

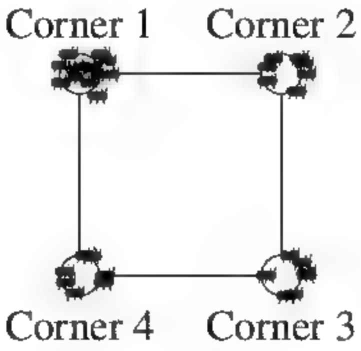
for u_1 and u_2 (where the coefficients -11 and 2 are the eigenvalues of M). ■

Remark 27.1.3. You may wonder how significant it is that in each of the two previous examples we arrived at 2×2 matrices that are symmetric. Our focus on examples with a symmetric matrix above was for expository simplicity (since the Spectral Theorem then ensured we could find an orthonormal basis of eigenvectors), to focus on the main point that eigenvalues are a powerful tool in the study of systems of differential equations. Many coupled systems of n "linear" differential equations have a governing $n \times n$ matrix that is not symmetric; does the above technique then break down?

The good news is that the mathematical technique based on eigenvalues works perfectly well beyond the symmetric case. However, the behavior can become considerably more subtle to analyze because beyond the symmetric case typically one has to account for the possibility of *complex numbers* as eigenvalues, and also the analogue of the matrix W will no longer be orthogonal (and there are some further issues related to the possibility of a repeated root of a characteristic polynomial). In the non-symmetric case discussed in Example 24.4.3 there was a full set of real eigenvalues but the orthogonality of W breaks down. You will learn how to handle such matters in Math 53.

27.2. Population dynamics. In Chapter 16 we saw that raising matrices to high powers often occurs when modeling what happens to a system over a long period of time. In Proposition 24.4.2 we stated a precise sense in which the eigenvalue that is largest (in the sense of absolute value) controls the behavior of high powers of a square matrix in the symmetric case, and in Example 24.4.3 we illustrated that the ideas carry over to the non-symmetric case. This application of eigenvalues is so important that we now use it to analyze yet another example, and then discuss a refined lesson concerning how a dominant eigenvalue controls high powers of a square matrix.

Example 27.2.1. Suppose there are several colonies of ants, one at each corner of a square. Initially, there are 10000 ants at one corner, and 1000 ants at each of the other three corners.



Each colony grows by 10% each week and at each corner there is also "diffusion": 20% of the original ant population (from the start of the week) leaves and moves clockwise to the next corner; another 20% of that original population leaves and moves counterclockwise to the next corner.

How many ants are there at each corner after 10 weeks? You might write a computer program to do this. However, the method we present below is, in general situations, *much more efficient* (and more illuminating). We encode the ant population at the end of week n as a vector giving the population at each corner of the square:

$$\mathbf{v}(n) = \begin{bmatrix} \text{population at corner 1 at the end of week } n \\ \text{population at corner 2 at the end of week } n \\ \text{population at corner 3 at the end of week } n \\ \text{population at corner 4 at the end of week } n \end{bmatrix},$$

so the initial population ("end of week 0") is encoded in the vector

$$\mathbf{P} = \mathbf{v}(0) = \begin{bmatrix} 10000 \\ 1000 \\ 1000 \\ 1000 \end{bmatrix}.$$

The description of what happens each week translates into the following matrix equation:

$$\mathbf{v}(n+1) = \underbrace{\begin{bmatrix} 0.7 & 0.2 & 0 & 0.2 \\ 0.2 & 0.7 & 0.2 & 0 \\ 0 & 0.2 & 0.7 & 0.2 \\ 0.2 & 0 & 0.2 & 0.7 \end{bmatrix}}_{B} \mathbf{v}(n) = B\,\mathbf{v}(n) \tag{27.2.1}$$

with B denoting the indicated 4×4 matrix. To explain this, let's consider what the first row is saying: the population at corner 1 at the end of week $n+1$ is equal to the sum of: 20% of population at corner 2 at the end of week n, 20% of the population at corner 4 at the end of week n, and $(1 - 0.2 - 0.2 + 0.1) = 0.7 = 70\%$ of the population at corner 1 at the end of week n. The 70% figure arises because 20% of the population at corner 1 at the end of week n is lost to diffusion into each of the two adjacent corners (so -0.2 twice) and there is also a 10% growth of the population at corner 1 at the end of week n (so $+0.1$).

Arguing as in Sections 16.1 and 16.2, feeding (27.2.1) into itself repeatedly yields $\mathbf{v}(n) = B^n\mathbf{v}(0) = B^n\mathbf{P}$ for all $n \geq 1$, so we want to compute $\mathbf{v}(10) = B^{10}\mathbf{P}$. We will exploit eigenvectors to compute this. The diffusion rates were chosen to make B symmetric, so the Spectral Theorem applies and hence there is an orthonormal basis of eigenvectors for B. (General diffusion problems involve high powers of a *non-symmetric* matrix, so then one has to grapple with issues as in Remark 27.1.3.) The matrix B has (exact) eigenvalues $\lambda_1 = 0.3$, $\lambda_2 = \lambda_3 = 0.7$, $\lambda_4 = 1.1$, with corresponding (unit) eigenvectors

$$\mathbf{w}_1 = \frac{1}{2}\begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix}, \mathbf{w}_2 = \frac{1}{2}\begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}, \mathbf{w}_3 = \frac{1}{2}\begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}, \mathbf{w}_4 = \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

(That is, $B\mathbf{w}_j = \lambda_j\mathbf{w}_j$ with $\|\mathbf{w}_j\| = 1$.) These nonzero vectors in $\mathbf{R}^4$ are pairwise orthogonal (check) and so constitute a basis of $\mathbf{R}^4$. This is an *orthonormal* basis since we arranged each $\mathbf{w}_j$ to be a unit vector.

Now write $\mathbf{P}$ as a linear combination of the members of the orthonormal basis $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4\}$ of $\mathbf{R}^4$ using the Fourier formula from Theorem 5.3.6 (with $\mathbf{v} = \mathbf{P}$ and $\mathbf{v}_i = \mathbf{w}_i$ there):

$$\mathbf{P} = \sum_{j=1}^{4}(\mathbf{P}\cdot\mathbf{w}_i)\mathbf{w}_i = -4500\mathbf{w}_1 - 4500\mathbf{w}_2 + 4500\mathbf{w}_3 + 6500\mathbf{w}_4. \tag{27.2.2}$$

The utility of this is that we can now easily compute what happens when we apply B ten times:

$$\begin{aligned} B^{10}\mathbf{P} &= -4500B^{10}\mathbf{w}_1 - 4500B^{10}\mathbf{w}_2 + 4500B^{10}\mathbf{w}_3 + 6500B^{10}\mathbf{w}_4 \\ &= -4500(0.3)^{10}\mathbf{w}_1 - 4500(0.7)^{10}\mathbf{w}_2 + 4500(0.7)^{10}\mathbf{w}_3 + 6500(1.1)^{10}\mathbf{w}_4, \end{aligned}$$

where the final equality expresses the fact that $B^{10}\mathbf{w}_j = \lambda_j^{10}\mathbf{w}_j$ for every j (a general feature of powers of a square matrix applied to an eigenvector of the matrix, as we saw in (24.3.1)).

This final expression for $B^{10}\mathbf{P}$ may look like a mess, but the key point is that the first three coefficients involve the 10th powers of the positive numbers 0.3 and 0.7 that are *less than* 1, and if we raise a number strictly between -1 and 1 to a high power then the output is *very close* to 0. Consequently, the first three coefficients in our final expression for $B^{10}\mathbf{P}$ are negligible, and we arrive at the good approximation

$$B^{10}\mathbf{P} \approx 6500(1.1)^{10}\mathbf{w}_4. \tag{27.2.3}$$

If we work everything out exactly on a computer, it turns out that

$$B^{10}\mathbf{P} = \begin{bmatrix} 8556 \\ 8430 \\ 8303 \\ 8430 \end{bmatrix},$$

and the curious feature is that the numbers are all rather close to each other. In other words, after 10 weeks the number of ants in the various corners has roughly equalized (in contrast with the substantial discrepancy in favor of corner 1 at the outset). To understand conceptually why this equalizing has occurred, the crucial point is that the entries of the eigenvector $\mathbf{w}_4 = \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ (with $\lambda_4 = 1.1$) are *all equal to each other* by inspection, so the same holds for *any* scalar multiple $c\mathbf{w}_4$, such as the right side of (27.2.3). The approximation gets even better as the number of weeks grows beyond 10, and the equalizing becomes even stronger. After a long time, all corners of the square have approximately the same number of ants.

In terms that apply beyond this setting (e.g., one might have several eigenvalues larger than 1), what matters is that $|\lambda_4|$ is larger than $|\lambda_1|, |\lambda_2|, |\lambda_3|$, so the ratios $\lambda_1/\lambda_4, \lambda_2/\lambda_4, \lambda_3/\lambda_4$ have absolute value less than 1. This is expressed by an important lesson in Proposition 24.4.2 that we repeat again:

> In situations involving raising a matrix to a power, *the eigenvalue that is largest* (in the sense of absolute value) *and a corresponding unit eigenvector* control the long-term behavior.

Eigenvalues tell us even more! If the second largest eigenvalue 0.7 had been very close to the largest eigenvalue λ_4 then we would have needed to pass to even larger n before the approximation $B^n\mathbf{P} \approx (\mathbf{P}\cdot\mathbf{w}_4)\lambda_4^n\mathbf{w}_4$ (replacing (27.2.3)) became valid. Stated more generally (but still informally):

> In situations involving raising a matrix to a power, the eigenvalue μ that is *second* largest (in absolute value) controls how *long* it takes before the eigenvalue λ that is largest (in absolute value) "takes over." The closer $|\mu|$ is to $|\lambda|$, the longer it takes for the behavior of λ and a corresponding unit eigenvector to dominate.

For further examples, go back to Chapter 16. Both Section 16.1 and Section 16.2 involve situations where, after many repetitions of a certain process (a bird migration and a coin flip respectively), the situation approached a "steady state". By a refinement of the statement of Theorem D.1.1, 1 is a dominant eigenvalue for each. If you compute (say on a computer) the eigenvalue μ whose absolute value is second largest, it is instructive to compare the ratio $1/|\mu|$ in each case with how quickly the system approaches its "steady state" in each case: for Section 16.1 we have $\mu = -1/2$ (so $1/|\mu| = 2$) and for Section 16.2 we have $\mu = (1+\sqrt{5})/4 \approx 0.8090$ (so $1/|\mu| = 1/\mu \approx 1.2361$). For the PageRank algorithm in Appendix D, $1/|\mu|$ is controlled by the parameter α; the choice $\alpha \approx 0.85$ thereby underlies why PageRank calculations finish in a matter of days and do not need vastly longer (if one tries α close to 1). ■

27.3. Singular value decomposition and principal component analysis. In Chapter 23, we discussed the fundamental Spectral Theorem, which says that for any symmetric $n \times n$ matrix A, there is an orthonormal basis $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ of $\mathbf{R}^n$ in terms of which the effect of A looks very simple: A simply "stretches" each $\mathbf{v}_i$ by a real number called the corresponding eigenvalue, which is to say $A\mathbf{v}_i = \lambda_i \mathbf{v}_i$ for some scalar λ_i (we called $\mathbf{v}_i$ an *eigenvector* of A, and λ_i the corresponding *eigenvalue*). Applications of eigenvalues and eigenvectors are pervasive throughout the natural sciences, data science, computer science, economics, statistics, and so on. These concepts are truly one of the most important things you learn in this course.

There are efficient and stable algorithms that compute the λ_i's and $\mathbf{v}_i$'s for a given symmetric matrix A, and we can use this to easily describe how A acts on any vector $\mathbf{v}$: writing $\mathbf{v} = c_1\mathbf{v}_1 + \cdots + c_n\mathbf{v}_n$ for scalars c_i given by the Fourier formula (5.3.5), we have

$$A\mathbf{v} = c_1 A\mathbf{v}_1 + \cdots + c_n A\mathbf{v}_n = c_1\lambda_1\mathbf{v}_1 + \cdots + c_n\lambda_n\mathbf{v}_n.$$

If Q is the matrix whose columns are the $\mathbf{v}_i$'s then Q is orthogonal and Theorem 24.4.1 (the matrix interpretation of the Spectral Theorem) gives that

$$A = QDQ^{\top} = QDQ^{-1} \tag{27.3.1}$$

where D is the diagonal matrix whose ith diagonal entry is the eigenvalue λ_i for A on $\mathbf{v}_i$.

Unfortunately, an orthogonal eigenvector decomposition of $\mathbf{R}^n$ to describe the effect of A (or equivalently a matrix factorization (27.3.1) with orthogonal Q and diagonal D) is only available when A is symmetric. It turns out that for a typical non-symmetric $n \times n$ matrix A, there is no basis of $\mathbf{R}^n$ consisting of eigenvectors. Moreover, even when such a basis exists, the algorithms for finding eigenvectors (for non-symmetric A) are numerically unstable. Furthermore, this circle of ideas only applies to square matrices, and in many applications one is confronted with $m \times n$ matrices with $m \neq n$.

There is a remarkable, and closely related, idea called the *singular value decomposition* (SVD). This makes up for all the pitfalls mentioned above: it works for **all** matrices A of any size and shape, and there are good stable algorithms to compute it. Furthermore, when A is a (square) symmetric matrix, the SVD literally is the matrix decomposition (27.3.1) up to some sign issues when there are negative eigenvalues. For this reason, SVD is regarded by almost everyone who uses applied linear algebra as **the** primary method for doing computations with a matrix of interest, and its importance in applications cannot be overstated (see [**MP**]). In this section we introduce the singular value decomposition and describe a few applications (especially "principal component analysis").

The basic idea is a remarkably simple tweak of what we have already done. If A is an $m \times n$ matrix, so the corresponding linear transformation $T_A : \mathbf{R}^n \to \mathbf{R}^m$ carries vectors $\mathbf{v} \in \mathbf{R}^n$ to vectors $A\mathbf{v} \in \mathbf{R}^m$, unless $m = n$ it is impossible to find a common basis in the source and target spaces (as the dimensions are not the same). Once we realize this, it is not a huge leap to wonder if we might find potentially quite different orthonormal bases for the input and output in terms of which the effect of A looks nice; even when $m = n$, this is a new idea!

To be precise, the main idea behind the SVD of a matrix A is that we look for an orthonormal basis $\mathbf{v}_1, \dots, \mathbf{v}_n$ of $\mathbf{R}^n$ and an orthonormal basis $\mathbf{w}_1, \dots, \mathbf{w}_m$ of $\mathbf{R}^m$ (which may be rather different from the $\mathbf{v}_i$'s when $m = n$!) so that for each $1 \le i \le n$

$$A\mathbf{v}_i = \sigma_i \mathbf{w}_i \tag{27.3.2}$$

for some scalar $\sigma_i \geq 0$. (This doesn't quite make sense if $m < n$, since $\mathbf{w}_i$ has no meaning when $i > m$, so in such cases we make the convention that $\mathbf{w}_i = \mathbf{0}$ for $i > m$.) In particular, $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ and $\{A\mathbf{v}_1, \ldots, A\mathbf{v}_n\}$ would *both* be orthogonal collections of vectors, which would be rather astounding to find for a general A; see Figure B.4.1 for why it is so surprising. The scalars σ_i are called the *singular values* of A; it is traditional to denote these by σ's, whereas eigenvalues are traditionally denoted by λ's.

Note that we require the $\mathbf{v}_i$'s and $\mathbf{w}_j$'s to be *orthonormal* bases, just like the basis of eigenvectors for a symmetric matrix in the Spectral Theorem (so if $m = n$ and A is not symmetric then this is a very different idea from an eigenvector basis for $\mathbf{R}^n$: if an $n \times n$ matrix A has eigenvectors giving a basis of $\mathbf{R}^n$, there never exists a basis of $\mathbf{R}^n$ consisting of pairwise *orthogonal* eigenvectors when A is not symmetric).

Remark 27.3.1. The reason for the notation σ for singular values is probably because this is the Greek version of "s", the initial letter in "singular value". In a special case of SVD called *principal component analysis* that will be discussed in Example 27.3.11, such σ's have a statistical interpretation as *standard deviations*. Hence, it is very convenient that standard deviations are denoted as σ in statistics, no doubt due to the coincidence that "standard deviation" also begins with the letter "s".

There are various ways to express the data in (27.3.2) (when it exists, as will turn out to always be the case). One common way is similar in spirit to the QR- and LU-decompositions in Chapter 22. To explain that, first we recall terminology from Definition 15.1.1 for general matrices (not assumed to be square).

Definition 27.3.2. An $m \times n$ matrix $D = (d_{ij})$ is called *diagonal* if the entries d_{ij} vanish whenever $i \neq j$; in other words, all "off-diagonal" entries vanish.

Here are two examples of diagonal $m \times n$ matrices with $m \neq n$:

$$\begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \\ 0 & 0 & 0 & \sigma_4 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \end{bmatrix}. \tag{27.3.3}$$

The first of these is a 5×4 matrix, and the second of these is a 3×4 matrix.

Here is the main result (proved in Section 27.4 using the Spectral Theorem):

Theorem 27.3.3 (Singular Value Decomposition (SVD)). For **every** $m \times n$ matrix A we can find:

- an $m \times n$ diagonal matrix D which has diagonal entries $d_{ii} = \sigma_i$, where $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$;
- an $m \times m$ orthogonal matrix Q and an $n \times n$ orthogonal matrix Q'

for which

$$A = QDQ'^{\top} \tag{27.3.4}$$

The diagonal matrix D is uniquely determined; the numbers σ_i are called the *singular values* of A.

Remark 27.3.4. In many references, one sees the right side of (27.3.4) written with different notation: $U\Sigma V^{\top}$ where U and V are orthogonal and Σ is diagonal with non-negative entries (arranged in decreasing order). The diagonal matrix is called Σ (Greek version of "S") rather than D because its diagonal entries (the "singular values") are denoted as σ_i. The reason Q is denoted as U (hence Q' is denoted by the next English letter V) in many references is not as easy to guess, so we now briefly explain the rationale.

If one permits the generality of matrices A with entries that are complex numbers (not necessarily real numbers) then in the analogue of the Spectral Theorem using complex numbers the role of orthogonal matrices is replaced by a wider class of square matrices called *unitary* (which coincides with "orthogonal" for matrices with entries in $\mathbf{R}$). This leads to the notation "U" if one wants to state the SVD result in a form that works for matrices with entries that are complex numbers. We have no reason to leave the familiar setting of real numbers, so we stick to the notation Q and Q' as above.

Remark 27.3.5. In the special case $m = n$ and A symmetric, the singular value decomposition is a formulation of the Spectral Theorem (Theorem 24.1.4) as given in (27.3.1). In that case, the orthogonal matrix Q has columns $\mathbf{w}_1, \dots, \mathbf{w}_n$ which are an orthonormal basis of eigenvectors for A, Q' is essentially the matrix Q except that its ith row is multiplied by -1 when the ith eigenvalue is negative, and the σ_i's are the *absolute values* of the eigenvalues λ_i of A. (The relevance of the negation in the definition of Q' is that if $A\mathbf{w} = \lambda\mathbf{w}$ with $\lambda < 0$ then $A\mathbf{w} = (-\lambda)(-\mathbf{w})$ with $-\lambda > 0$.)

In general, if $m = n$ but A is not symmetric then the singular values σ_i are *not* eigenvalues of A (or even their negatives). Of course, if $m \neq n$ then the notion of eigenvalue for A doesn't even make any sense, since an equation such as $A\mathbf{v} = \lambda\mathbf{v}$ has no meaning if $m \neq n$ (the left side is a vector in $\mathbf{R}^m$ and the right side is a vector in $\mathbf{R}^n$, since A is applied to input from $\mathbf{R}^n$ and yields output in $\mathbf{R}^m$). However, the SVD is still intimately intertwined with eigenvalues, and the proof of the Theorem 27.3.3 deduces SVD for any A from the Spectral Theorem for the symmetric $n \times n$ matrix $A^\top A$ (a positive-semidefinite matrix, as discussed in Example 26.1.10).

Let us now give a couple of examples of SVD's and then turn to a few applications.

Example 27.3.6. We first consider an example where $m < n$. Let

$$A = \begin{bmatrix} 2 & 1 & 1 \\ -2 & 1 & 3 \end{bmatrix}.$$

Here $m = 2$ and $n = 3$. The singular value decomposition $A = QDQ'^\top$ takes the form

$$\begin{bmatrix} 2 & 1 & 1 \\ -2 & 1 & 3 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \sqrt{14} & 0 & 0 \\ 0 & \sqrt{6} & 0 \end{bmatrix} \begin{bmatrix} -\sqrt{2/7} & 1/\sqrt{14} & 3/\sqrt{14} \\ \sqrt{2/3} & 1/\sqrt{6} & 1/\sqrt{6} \\ 1/\sqrt{21} & -4/\sqrt{21} & 2/\sqrt{21} \end{bmatrix}.$$

Following the discussion above, this SVD means that if we define the vectors $\mathbf{w}_1, \mathbf{w}_2$ to be the columns of Q and $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ to be the columns of Q' (so the *rows* of $Q'^\top$!), then

$$A\mathbf{v}_1 = \sqrt{14}\,\mathbf{w}_1, \quad A\mathbf{v}_2 = \sqrt{6}\,\mathbf{w}_2, \quad A\mathbf{v}_3 = \mathbf{0}.$$

■

Example 27.3.7. Next we consider an example where $m > n$; now we will take $m = 3$ and $n = 2$ and set

$$A = \begin{bmatrix} 2 & -1 \\ -1 & 3 \\ -1 & 1 \end{bmatrix}.$$

This has singular value decomposition

$$\begin{bmatrix} 2 & -1 \\ -1 & 3 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} -7/\sqrt{195} & 2\sqrt{2/13} & \sqrt{2/15} \\ 11/\sqrt{195} & 3/\sqrt{26} & -1/\sqrt{30} \\ \sqrt{5/39} & -1/\sqrt{26} & \sqrt{5/6} \end{bmatrix} \begin{bmatrix} \sqrt{15} & 0 \\ 0 & \sqrt{2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -2/\sqrt{13} & 3/\sqrt{13} \\ 3/\sqrt{13} & 2/\sqrt{13} \end{bmatrix}.$$

Denoting the columns of Q as $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3$ and the columns of Q' (so the *rows* of $Q'^\top$!) as $\mathbf{v}_1, \mathbf{v}_2$, we have $A\mathbf{v}_1 = \sqrt{15}\,\mathbf{w}_1$, $A\mathbf{v}_2 = \sqrt{2}\,\mathbf{w}_2$ (and $\mathbf{w}_3$ is orthogonal to $\mathrm{span}(\mathbf{w}_1, \mathbf{w}_2) = \mathrm{image}(A)$). ■

You may wonder how these SVD computations were done. There is a lot of software to compute such decompositions. For matrices of moderate size, Wolfram can compute them using the command "SingularValueDecomposition". Much as the SVD for a symmetric matrix is closely related to its eigenvector decomposition, SVD for a general matrix A (possibly not symmetric, nor even square) is also related to eigenvectors. Indeed, from the proof of Theorem 27.3.3 the $\mathbf{v}_i$'s are eigenvectors of the symmetric $A^\top A$ and the $\mathbf{w}_i$'s are eigenvectors of the symmetric $AA^\top$, and the *squares* σ_i^2 of the nonzero singular values $\sigma_i > 0$ of A are eigenvalues of these symmetric matrices! Thus, computing eigenvalues and eigenvectors for symmetric square matrices is nearly enough to compute SVD's in general.

To be precise, if $m \leq n$ then $AA^\top$ is an $m \times m$ matrix and $A^\top A$ is an $n \times n$ matrix. But every matrix of the form $M^\top M$ (e.g., using $M = A$ or $M = A^\top$) is not only symmetric but also positive-semidefinite since $\mathbf{x}^\top M^\top M\mathbf{x} = (M\mathbf{x})^\top(M\mathbf{x}) = (M\mathbf{x}) \cdot (M\mathbf{x}) = \|M\mathbf{x}\|^2 \geq 0$. The Spectral Theorem (and Proposition 24.2.10) applied to any $M^\top M$ thereby yields eigenvalues that are non-negative and hence squares in $\mathbf{R}$. The eigenvalues of $AA^\top$ can therefore be written as $\sigma_1^2, \dots, \sigma_m^2$, and the eigenvalues of $A^\top A$ consist of the same list augmented by $n - m$ additional 0's. (When $m \geq n$, the situation is analogous.) Here we have invoked the remarkable fact that the *nonzero* eigenvalues of $AA^\top$ and $A^\top A$ are always the same. More generally, if A is an $m \times n$ matrix and B is an $n \times m$ matrix with $m \leq n$ then the eigenvalues BA are exactly those of AB along with $n - m$ additional occurrences of 0 as an eigenvalue; we omit the proof (which amounts to an elegant application of the notion of characteristic polynomial introduced for 2×2 matrices in Theorem 23.3.1 and in general in Theorem E.5.1 and Remark E.5.2).

We now turn to some fundamental practical applications of SVD.

Example 27.3.8 (Rank reduction). The following problem is highly relevant for data compression.

Suppose that you have an $m \times n$ matrix A where m and n are on the order of 10^6, where the individual entries a_{ij} of A correspond to some data that you have collected. For example, these entries might encode light intensity and/or color at the points of an $m \times n$ grid which can be reassembled into a digital approximation of an image. It takes a lot of memory to store and possibly manipulate $mn \approx 10^{12}$ numbers. Is there a way to avoid such storage problems?

The basic idea of data compression is that it would be very nice to somehow reduce this information to a much smaller set of numbers which still captures almost all of the features of the original data set (which itself is perhaps only a sampling of the true image). This sounds like a fantasy, but remarkably there are a number of very ingenious ideas which actually do it very effectively. Variants of these ideas are behind the jpg and mpg formats, and are responsible for some of the most dramatic advances in medical imaging (e.g. MRI). We now explain the key underlying mathematical idea.

To motivate this, let us examine another way of interpreting the singular value decomposition $QDQ'^\top$ of an $m \times n$ matrix A. Suppose we write the columns of the "prefactor" orthogonal $m \times m$ matrix Q as $\mathbf{w}_1, \dots, \mathbf{w}_m$ and the columns of the "postfactor" orthogonal $n \times n$ matrix Q' as $\mathbf{v}_1, \dots, \mathbf{v}_n$. Then we can multiply out the terms in

$$A = QDQ'^\top = \begin{bmatrix} | & | & & | \\ \mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_m \\ | & | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \sigma_m & 0 \end{bmatrix} \begin{bmatrix} — & \mathbf{v}_1^\top & — \\ — & \mathbf{v}_2^\top & — \\ & \vdots & \\ — & \mathbf{v}_n^\top & — \end{bmatrix}$$

(here we are illustrating the case $m \leq n$ to be specific) as

$$A = \sigma_1 \mathbf{w}_1 \mathbf{v}_1^\top + \cdots + \sigma_m \mathbf{w}_m \mathbf{v}_m^\top. \tag{27.3.5}$$

Note that the vectors $\mathbf{v}_{m+1}, \dots, \mathbf{v}_n \in \mathbf{R}^n$ do not appear here. For similar reasons, if $m > n$ then this sum has n terms and the vectors $\mathbf{w}_{n+1}, \dots, \mathbf{w}_m \in \mathbf{R}^m$ do not appear.

What are these funny products $\mathbf{w}_i \mathbf{v}_i^\top$ that appear here, which are $m \times n$ matrices? These are instances of what are called *rank*-1 *matrices*. You might try to understand them by writing out the entries: again supposing $m \leq n$, we have

$$\mathbf{w}_i = \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix}, \; \mathbf{v}_i = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \text{ implies } \mathbf{w}_i\mathbf{v}_i^\top = \begin{bmatrix} a_1b_1 & a_1b_2 & \dots & a_1b_n \\ a_2b_1 & a_2b_2 & \dots & a_2b_n \\ \vdots & \vdots & \vdots & \vdots \\ a_mb_1 & a_mb_2 & \dots & a_mb_n \end{bmatrix}.$$

That matrix might not be particularly illuminating at first sight. The special feature it has is that *every* column is a multiple of the *same* vector $\mathbf{w}_i$ (the jth column is $b_j\mathbf{w}_i$). More concretely, let's calculate the effect of this matrix on any vector $\mathbf{x}$ via the matrix-vector product: it is simply $(\mathbf{w}_i\mathbf{v}_i^\top)\mathbf{x} = \mathbf{w}_i(\mathbf{v}_i^\top\mathbf{x})$, but $\mathbf{v}^\top\mathbf{x}$ is a 1×1 matrix, which is to say a *scalar*. This scalar is just the dot product $\mathbf{v}_i \cdot \mathbf{x}$, so

$$(\mathbf{w}_i\mathbf{v}_i^\top)\mathbf{x} = (\mathbf{v}_i \cdot \mathbf{x})\mathbf{w}_i.$$

(The fact that the linear transformation arising from the $m \times n$ matrix $\mathbf{w}_i\mathbf{v}_i^\top$ has all output contained in a 1-dimensional subspace of $\mathbf{R}^m$ is the reason it is called a "rank-1 matrix".) Thus,

$$A\mathbf{x} = \left(\sigma_1\mathbf{w}_1\mathbf{v}_1^\top + \cdots + \sigma_m\mathbf{w}_m\mathbf{v}_m^\top\right)\mathbf{x} = (\sigma_1\mathbf{v}_1 \cdot \mathbf{x})\mathbf{w}_1 + \cdots + (\sigma_m\mathbf{v}_m \cdot \mathbf{x})\mathbf{w}_m.$$

Since the scalar $\mathbf{v}_i \cdot \mathbf{x}$ is the length of the orthogonal projection of $\mathbf{x}$ onto the *unit* vector $\mathbf{v}_i$, we are simply taking the lengths of the orthogonal projections of $\mathbf{x}$ along the m orthonormal unit vectors $\mathbf{v}_1, \dots, \mathbf{v}_m$ in $\mathbf{R}^n$ (these can be thought of as "coordinates" of the n-vector $\mathbf{x}$ in those m directions), multiplying these lengths by the scalars σ_i, and then using these as coefficients of an expansion in terms of the orthonormal basis $\mathbf{w}_1, \dots, \mathbf{w}_m$ of $\mathbf{R}^m$.

Suppose that only one of these terms appears in the SVD of A, or in other words, suppose that $\sigma_2 = \ldots = \sigma_m = 0$ (again assuming $m \leq n$ for simplicity). Then we are writing

$$A = \sigma_1\mathbf{w}_1\mathbf{v}_1^\top.$$

How much "data" , or in other words how many numbers, are on the right side? There are only $m + n + 1$ numbers: the coordinates of $\mathbf{w}_1$ and $\mathbf{v}_1$ and the number σ_1. In other words, in such cases we have encoded the mn entries of A using only $m+n+1$ numbers. So if m and n are approximately 10^9, this is a huge gain: we only need to store around 10^9 numbers rather than 10^{18}.

It might seem absurdly special to consider situations in which all but the first singular value vanish. But the miracle of the technique called *rank reduction* is that something not too far off from this often happens: for many matrices A which *arise in practice*, most of the singular values are relatively or very small! For example, if the singular values $\sigma_{k+1}, \dots, \sigma_m$ which appear in (27.3.5) are all quite small then (since $\mathbf{v}_i$ and $\mathbf{w}_i$ are unit vectors, so their entries cannot be too large) it is reasonable to just drop these to get a good approximation:

$$A = \sigma_1\mathbf{w}_1\mathbf{v}_1^\top + \cdots + \sigma_m\mathbf{w}_m\mathbf{v}_m^\top \approx \sigma_1\mathbf{w}_1\mathbf{v}_1^\top + \cdots + \sigma_k\mathbf{w}_k\mathbf{v}_k^\top. \tag{27.3.6}$$

We are taking advantage of the convention we imposed that we list the singular values in *decreasing* order, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$; we only need to look at the later part of this list to see which are the small singular values.

This can lead to a very good approximation of A by a sum of a possibly much smaller number of rank-1 matrices. For example, suppose that $k = 3$ (i.e., all singular values past the third one are negligible) and (once again) $m, n \approx 10^9$. Then we are finding a good approximation to all the $mn = 10^{18}$ entries of A using only around 3×10^9 numbers. (See Remark 27.3.9 for an actual situation in genetics with $k = 3$ and m on the order of 1000's.)

How good of an approximation is this? If $\sigma_{k+1}, \dots, \sigma_m$ are all less than some very small number ε, then in a certain very good sense we are approximating every entry of A to within ε. This type of quantitative information is very useful: we have a good approximation of A *and* an explicit numerical estimate of just how good an approximation it is! ■

Remark 27.3.9 (Eigengenes). In the paper [ABB], SVD was applied to a 5981×14 matrix and a 4579×22 matrix (respectively encoding the outcomes of measurements of 5981 genes under 14 different experimental conditions and of 4579 genes under 22 different experimental conditions, each in the context of yeast genetics). In both cases, the singular values dropped off dramatically after the first few. This work introduced the concept of "eigengene" as a mathematical tool for studying gene expression with useful biological meaning. For such $m \times n$ matrices A encoding genetic expression measurements, the eigengenes correspond to $\mathbf{v}_1, \mathbf{v}_2, \dots \in \mathbf{R}^m$ in the SVD for A. Orthogonality among the $\mathbf{v}_i$'s provides a sense in which different eigengenes are "uncorrelated" with each other.

Remark 27.3.10. There is a practical issue relating back to our work with optimization techniques (and so going far beyond linear algebra): does identifying the biggest among the singular values σ_j (and corresponding $\mathbf{w}_j$'s) require computing *all* singular values and then selecting the biggest? If so, that would be a computationally hopeless task (for big m and n) which would render rank-reduction useless.

Amazingly, finding the biggest singular value σ_1 (and corresponding $\mathbf{v}_1$ and $\mathbf{w}_1$) does *not* require computing any of the other singular values! One can use the Spectral Theorem to formulate the problem of finding the biggest singular value as an optimization problem that can be efficiently solved by ideas related to the technique of *gradient descent* which we explored at a basic level in Chapter 11 and study a bit more deeply via Newton's method based on second derivatives in Appendix I. The optimization problem even appears in the *proof* of the Spectral Theorem; see Remark B.3.4 and then Remark B.3.3.

Example 27.3.11 (Principal component analysis). A famous application of the SVD method is called *principal component analysis*, which is often referred to by the shorthand label "PCA". This is a fundamental tool in statistics, and one could write an entire book about this topic alone. When A is a matrix whose columns are labeled by members of a population and rows are categories in which they're measured (these categories could encode genetic information, movie ratings, etc.), the $\mathbf{w}_j$'s are called the *principal components* of the data. For any data m-vector $\mathbf{x}$, we call the coefficient $\mathbf{w}_j \cdot \mathbf{x}$ of the projection of $\mathbf{x}$ along the line spanned by $\mathbf{w}_j$ a "principal component" of $\mathbf{x}$; the standard coordinates of $\mathbf{x}$ are what we initially measure when collecting data, but their linear combinations given by the principal components $\mathbf{w}_j \cdot \mathbf{x}$ are regarded as deeper structure in the data (which may or may not have a tangible meaning).

The idea that most singular values are negligible, yielding an approximation to A as in (27.3.6) for some small k, corresponds to saying that the column space $C(A)$ is "close" to a subspace W of $\mathbf{R}^m$ spanned by the vectors $\mathbf{w}_1, \dots, \mathbf{w}_k$ corresponding to the non-negligible singular values. This underlies how one can make predictions about any person's entire m-vector $\mathbf{x}$ of measurements upon making a small number of measurements (around $k = \dim W$), as is discussed below in some real-world situations.

We shall now briefly convey the point that PCA is fundamentally just a reinterpretation of SVD, and more specifically of the ideas behind rank reduction discussed in Example 27.3.8. This may not be so obvious if you read many treatments of PCA, since the method and ideas are usually couched in the language of statistics (which is relevant for many applications, but can sometimes make it hard to see that the mathematical input is really SVD). In quantitative finance, PCA is used for purposes such as risk management, option pricing, and filling in missing data (see [Al, I.2.6] for an introduction).

You should regard PCA and SVD as a great illustration of how the same mathematical idea can have two rather different interpretations and applications (such as: one related to data compression and the other to statistical correlation). This is a manifestation of the unifying power of mathematics.

Here are some typical situations where PCA arises:

(i) Data compression, especially with images, is a dramatic illustration of the power of PCA. This also underlies facial and handwriting recognition software, and is increasingly used for both efficiency and fraud detection in the insurance industry (see [**Shang**, pp. 13-17] and this). Consider a digital photo that is a 300×300 pixel array, suppose in black-and-white (for convenience of discussion). The intensity of darkness at each pixel is a number from 0 (white) to 1 (black). In a real-world image, adjacent pixels are rather highly correlated: an image is far from a totally random assignment of pixel intensities. To exploit this feature of a real-world image, we break up the 300×300 array of numbers into smaller patches of 10×10 pixel arrays (for example). The part of the image in each patch is recorded by 100 pixel intensities, so a vector in $\mathbf{R}^{100}$. There are 900 of these vectors, so they can be arranged as the columns in a 100×900 matrix A.

The SVD of A has singular values σ_i that become negligible by around σ_{50}; i.e., we can usually take k as in (27.3.6) to be around 40 to 50. Thus, by recording the principal components $\sigma_1\mathbf{w}_1, \dots, \sigma_{50}\mathbf{w}_{50} \in \mathbf{R}^{100}$ and the vectors $\mathbf{v}_1, \dots, \mathbf{v}_{50} \in \mathbf{R}^{900}$ we can reconstruct an excellent approximation to A (as $\sum_{i=1}^{50}(\sigma_i\mathbf{w}_i)\mathbf{v}_i^\top$). This amounts to 50 vectors in $\mathbf{R}^{100}$ and 50 vectors in $\mathbf{R}^{900}$, which is $50(100) + 50(900) = 50,000$ numbers, whereas the original image has $90,000$ pixel intensities. So we have compressed the data to $5/9 \approx 55\%$ of its original size! If you search on Google for "image compression, PCA" then you'll find many dramatic examples of the accuracy of considering 30, 40, and 50 principal components for digital images.

(ii) The idea that applying PCA techniques to human DNA could lead to new insights in population genetics goes back to the 1970's, as documented in the landmark book [**CMP**]. But in more recent years, thanks to advances in computer power and genotyping technology, there has been an explosion in applications of PCA to human genetics (see [**N**] for an especially famous example). When analyzing the genome of a person (such as to make predictions about susceptibility to a specific genetic disease, or the likely response to a potential treatment), scientists measure a large collection of what are called *single nucleotide polymorphisms* (abbreviated as: SNP's). This refers to looking at specific locations in the DNA of each person, and recording which of various possibilities ("polymorphisms") occur at each location for the molecular structure of the DNA ("single nucleotide"). The ability to make such precise measurements is a remarkable accomplishment of modern biological engineering.

The possible variants found for the genetic code at each location may be labeled by one of a collection of numbers, say $1, 2, 3, 4, 5$ (if there are up to 5 possible genetic variants at each location). If a study records the SNP's at m locations in the DNA molecule for n people (e.g., 1000 sites in the genome for each of 10000 people), then all the data are encoded as the entries of an $m \times n$ matrix. Here the jth column records the genetic information of the jth person's DNA at the locations that have been measured.

One can ask if there is a significant correlation between the SNP found at one location in the DNA molecule and the SNP found at another location. For example: is a specific genetic expression found at one location in the DNA correlated well with what is found at another location (in which case one might arrive at a new way of testing for a genetic disease)? Even though the n column vectors (each of which is the genetic information of a single person) all live in $\mathbf{R}^m$ for $m \approx 1000$, it might be the case that those vectors all lie *very close to a very low-dimensional subspace* of $\mathbf{R}^m$. In that case each vector is well-approximated by its orthogonal projection into that low-dimensional subspace, and the low-dimensionality tells us that knowledge of a moderate amount of DNA information can make accurate predictions about the rest!

(iii) Another situation arises in the sneaky ways that companies keep track of consumer behavior. For example, a company might record the purchasing habits of n people among m items, and record the number of times that consumer j purchased item i. That number is then recorded as the ij-entry in an $m \times n$ matrix (so the jth column records the purchasing habits of the jth person, at least for the collection of m items of interest to the company).

If the company is Amazon and the items are all books sold by Amazon, m could be truly gigantic. Likewise, the Netflix and Spotify recommendation systems have to grapple with rather huge m. If the company is a home furnishing store and the items are its entire inventory then m won't be as large as Amazon's collection of books or Netflix's library of movies and TV shows or Spotify's list of songs, but it can still be pretty big.

The value in understanding the correlations between the matrix entries is to create predictive targeted advertising. As with the DNA example, each column lives in $\mathbf{R}^m$ for some large m, but the columns may all be close to a lower-dimensional subspace $V \subset \mathbf{R}^m$. In the context of consumer behavior, this low-rank behavior reflects the fact that there don't tend to be a huge number of ways in which personal preferences vary. The significance is that it enables knowledge about a small set of purchases (on the order of $\dim V$) to accurately predict a customer's preferences related to *other* items that they might purchase. (See Example 21.6.3 for more on this.)

Here is how such problems are analyzed. Given an $m \times n$ matrix A, let us call the columns of this matrix $\mathbf{X}_1, \ldots, \mathbf{X}_n$, so each $\mathbf{X}_j$ is a vector in $\mathbf{R}^m$. In both of the above examples, $\mathbf{X}_j$ records the information about the jth person. For instance, in the DNA example above, each entry of $\mathbf{X}_j$ might be one of the numbers $1, 2, 3, 4, 5$; the actual entries don't matter too much, as long as they are numbers. The first step is to compute the *mean* (or average) $\mathbf{M} = (\mathbf{X}_1 + \cdots + \mathbf{X}_n)/n$ of the vectors $\mathbf{X}_j \in \mathbf{R}^m$ and to replace each $\mathbf{X}_j$ by $\mathbf{Y}_j = \mathbf{X}_j - \mathbf{M}$; this has the effect of "recentering" all of the data much as in our study of correlation coefficients in Chapter 2. Let us call by the name B the $m \times n$ matrix with these new columns $\mathbf{Y}_1, \ldots, \mathbf{Y}_n$.

Now perform the SVD decomposition $B = QDQ'^{\top}$, and write this as before:

$$B = \sigma_1 \mathbf{w}_1 \mathbf{v}_1^{\top} + \sigma_2 \mathbf{w}_2 \mathbf{v}_2^{\top} + \ldots$$

with $\sigma_1 \geq \sigma_2 \geq \ldots$ and pairwise orthogonal unit vectors $\mathbf{w}_1, \mathbf{w}_2, \cdots \in \mathbf{R}^m$ and $\mathbf{v}_1, \mathbf{v}_2, \cdots \in \mathbf{R}^n$. It is very often the case in real-world data that one or a few of the σ_j are significantly larger than all of the others; this is the same phenomenon encountered in the rank-reduction discussion in Example 27.3.8.

Suppose for example that the biggest singular value σ_1 is much larger than all the other singular values. This means that a good approximation to B is the rank-1 matrix $\sigma_1 \mathbf{w}_1 \mathbf{v}_1^{\top}$. As we saw in Example 27.3.8, this means that every column is approximately a scalar multiple of the *same* vector $\mathbf{w}_1$. Tracing back through the steps, the conclusion is that each of the original vectors $\mathbf{X}_j$ is very well-approximated by $\mathbf{M} + c_j \mathbf{w}_1$ for some scalar multiplier c_j against a *single* unit vector $\mathbf{w}_1$. This is precisely what it means, at least informally, for the entries of $\mathbf{X}_j$ to be correlated! The direction $\mathbf{w}_1$ is then called the *principal component* of this collection of vectors, and once we recenter the vectors by subtracting off their collective mean vector, each becomes approximately positioned along this axis.

In concrete terms, to say that the m-vector $\mathbf{Y}_j$ is approximately $c_j \mathbf{w}_1$ for some (nonzero) scalar c_j and a fixed unit vector $\mathbf{w}_1$ is to say that for any $1 \leq i < i' \leq m$ the ratio of the ith and i'th entries in $\mathbf{Y}_j$ is approximately *independent* of j. Indeed, if we write w_k to denote the kth entry in $\mathbf{w}_1$ (and we assume w_k's are all nonzero) and we write a_{jk} to denote the kth entry in $\mathbf{Y}_j$ then $a_{jk} \approx c_j w_k$, so

$$\frac{a_{ji}}{a_{ji'}} \approx \frac{c_j w_i}{c_j w_{i'}} = \frac{w_i}{w_{i'}}$$

with the right side independent of j. This ratio is an approximate "slope" that tells us how to pass between the i'th and ith entries of *every* $\mathbf{Y}_j$.

As we said at the outset, this is only the barest introduction to the idea behind PCA; this method for analyzing data sets has many subtleties, and to use it effectively one really has to do much more study to come to grips with the subtleties. For example: how should we interpret the outcome of the SVD for B if two or three of the singular values σ_j (rather than just one of them) are much larger than the others? A proper answer involves a careful discussion of what correlation means in statistics, a topic that would take us too far afield in this course. Hopefully when you encounter this set of ideas again, the preceding orientation will help you to see how PCA relies in an essential way on core concepts and results you have learned about in linear algebra (in order to define and work with singular values) and optimization (in order to actually compute the biggest singular values). ■

27.4. Where does SVD come from? We are going to use the Spectral Theorem (Theorem 24.1.4), a serious result about eigenvectors, to explain where the singular value decomposition comes from by sketching a proof of Theorem 27.3.3. The argument below is an approach via matrix algebra, for which we will gloss over some technical points. An alternative and more complete geometric argument is given in Section B.4. The interested reader is welcome to study whichever approach fits more to their mathematical taste. Here is a proof based on matrix algebra:

PROOF. For purposes of motivation, let us first *suppose* that we already had a decomposition $A = QDQ'^{\top}$ with a diagonal $m \times n$ matrix D and orthogonal Q and Q'. How would we go about finding Q and Q' from A? The trick is to consider $A^{\top}A$. We have $A^{\top} = Q'D^{\top}Q^{\top}$, and since $Q^{\top}Q = \mathrm{I}_m$ (as Q is orthogonal) we obtain

$$A^{\top}A = Q'(D^{\top}D)Q'^{\top} \tag{27.4.1}$$

If the diagonal $m \times n$ matrix D has ii-entry λ_i then $D^{\top}D$ is a diagonal $n \times n$ matrix with ii-entry λ_i^2 for $i \leq \min(m, n)$ (and vanishing otherwise): this is familiar from matrix algebra when $m = n$, and the cases $m > n$ and $m < n$ can be checked directly in the specific cases $(m, n) = (5, 4)$ and $(m, n) = (3, 4)$ for D as in (27.3.3), with the general cases for $m < n$ and $m > n$ going exactly the same way.

In other words, for the $n \times n$ matrix $A^{\top}A$ that is always *symmetric* (see Theorem 20.3.8), (27.4.1) expresses the conclusion of the Spectral Theorem with the orthogonal Q' and the diagonal $n \times n$ matrix $D^{\top}D$ whose entries are the eigenvalues of $A^{\top}A$.

With the preceding considerations as *motivation* (since they relied on *assuming* the existence of the D, Q, Q' we seek to find), we are now motivated to know where to look to find D, Q, and Q'. Apply the Spectral Theorem to the symmetric $n \times n$ matrix $A^{\top}A$ to get an orthogonal $n \times n$ matrix Q' (this has as its columns an orthonormal basis of eigenvectors $\mathbf{w}_1, \dots, \mathbf{w}_n$ of $A^{\top}A$) and a diagonal $n \times n$ matrix D' (whose ith diagonal entry is the $A^{\top}A$-eigenvalue for $\mathbf{w}_i$) satisfying

$$A^{\top}A = Q'D'Q'^{\top}.$$

We want to write $D' = D^{\top}D$ for a diagonal $m \times n$ matrix D. Getting such a D requires choosing square roots of the entries of D', so there is something to be careful about here: picking such square roots requires knowing that every eigenvalue of $A^{\top}A$ (that is, the diagonal entries of D') is ≥ 0. But this is indeed the case, by Example 26.1.10 and Proposition 24.2.10 (applied to the symmetric $A^{\top}A$).

Some care is required to choose the signs of the square roots appropriately; we will gloss over that issue here, and now focus on the special case $m = n$ with A invertible (so its eigenvalues are all nonzero, hence D' is also invertible and thus its "square root" D is invertible). In such cases we can find the remaining matrix Q using inverses: starting from the desired relation $A = QDQ'^{\top}$ we are motivated to define the $m \times m$ matrix $Q = AQ'D^{-1}$. By design $QDQ'^{\top} = AQ'D^{-1}DQ'^{\top} = AQ'Q'^{\top} = A\mathrm{I}_n = A$, so we just have to check that Q is orthogonal; i.e., to show $Q^{\top}Q = \mathrm{I}_m$.

Using the interaction of transpose with matrix multiplication and matrix inversion (Section 20.1 and Example 20.3.7), by definition of Q we have

$$Q^\top = (AQ'D^{-1})^\top = (D^{-1})^\top Q'^\top A^\top = (D^\top)^{-1} Q'^\top A^\top = D^{-1} Q'^\top A^\top,$$

the final step using that $D^\top = D$ (as we are in the special case $n = m$). Thus,

$$\begin{aligned} Q^\top Q = D^{-1} Q'^\top A^\top (AQ'D^{-1}) = D^{-1} Q'^\top (A^\top A) Q' D^{-1} &= D^{-1} Q'^\top (Q'D'Q'^\top) Q' D^{-1} \\ &= D^{-1} (Q'^\top Q') D' (Q'^\top Q') D^{-1}. \end{aligned}$$

But Q' and $Q'^\top$ are inverse to each other since Q' is orthogonal, so $Q^\top Q = D^{-1} D' D^{-1}$. But D^{-1} and D' are diagonal $m \times m$ matrices, and multiplication of such diagonal matrices is *commutative* (i.e., the order of multiplication doesn't matter), so since $D' = D^2$ we can rearrange to get

$$Q^\top Q = D'(D^{-1})^2 = D^2 (D^{-1})^2 = DDD^{-1}D^{-1} = DD^{-1} = \mathrm{I}_m.$$

This establishes the orthogonality of Q as desired (assuming $m - n$ and A is invertible).

In the general case (allowing A not to be invertible and even $m \neq n$) one can still solve for Q by a variant of the preceding method, but it's a little more complicated to explain, so we stop the discussion of the matrix-algebra proof of SVD here. (Once again, Section B.4 provides a more complete argument based on geometric ideas.) ☐

Appendices

“The trigonometric functions $\sec x$ and $\csc x$ do not exist in France, so I will not use them.”

H. Cohen [Coh, p. ix]

A. Review of functions

This appendix discusses some general topics concerning functions, as a convenient resource. Some of this material may have arisen in your earlier math courses; we discuss it here to ensure that we are all using the same terminology and notation for basic concepts.

A.1. Several basic examples. Functions are mathematical tools for describing the world around us. A wide variety of functions arise in calculus and in real-world problems. Here are some examples.

Example A.1.1. In calculus courses you probably saw examples of functions such as $f(x) = 3x/(x-1)$. Here the variable x is a real number *input*, and the expression on the right side gives us a formula or *rule* for finding the *output* of the function corresponding to the input x. The output is denoted $f(x)$.

Calculus is described as the study of change. When analyzing the function above in a calculus course, we might ask questions such as:

(i) How do small changes in x affect $f(x)$?
(ii) How is $f(x)$ changing near some point $x = a$? Is it increasing? Decreasing?

In calculus one learns how to compute and use derivatives to analyze such questions (and do much more). ■

Example A.1.2. A microbiologist is working with a previously undescribed bacterial organism. Via experimentation, she discovers that the population of the bacterium doubles every 13.4 hours, assuming unlimited resources (such as food and space). Under these idealized assumptions, she proposes a first-guess estimate for the population function of the bacteria: $P(t) = P(0) \cdot 2^{t/13.4}$, where $P(0)$ is the initial population and t is time elapsed (in hours).

In this mathematical model for population growth, the only dependence is on $P(0)$ and t. But as her experiments progress, the microbiologist may use her data to refine the model by taking into account other variables upon which the population growth is discovered to depend. Other than the initial population in the sample and the time elapsed, some variables that might affect the bacteria's population growth could include the salinity of the water in which the bacteria live, the temperature of the room, and so on. ■

Example A.1.3. Isaac Newton's theory of gravitation asserts that the force of gravity between two objects (with masses m_1 and m_2) is given by $F(m_1, m_2, r) = (Gm_1m_2)/r^2$, where G is a constant and $r > 0$ is the distance between the two objects. This function has three pieces of information as its input: the masses of the two objects and the distance between them. We can therefore think of the inputs to F as being points (m_1, m_2, r) in $\mathbf{R}^3$ (ordered triples of real numbers) with $r > 0$.

In this mathematical model, if we hold m_1 and m_2 constant but increase r then $F(m_1, m_2, r)$ *decreases*; more specifically, if we double (or triple) the distance then F drops by a factor of 4 (or a factor of 9 respectively). Informally, one hears it said that “F is proportional to the inverse square of the distance”, but the mathematical formula is a more precise way of conveying the nature of F (encoding that the constant of proportionality is Gm_1m_2, which the verbal description does not provide). Likewise, if we hold m_2 and r constant but increase m_1 then $F(m_1, m_2, r)$ increases proportionally (with constant of proportionality Gm_2/r^2). ■

Example A.1.4. A meteorologist is studying wind patterns in some region R at a particular point in time. He models wind as a function $\mathbf{w}$ where, for a given point $\mathbf{x}$ in R one defined

$$\mathbf{w}(\mathbf{x}) = (\text{wind direction at } \mathbf{x}, \text{wind speed at } \mathbf{x}).$$

The output $\mathbf{w}(\mathbf{x})$ consists of two components: wind direction (as an angle in degrees, say in the interval $[0, 360)$) and wind speed (in miles per hour). Thus, we may think of $\mathbf{w}$ as a function taking inputs in the set R and giving outputs in the set $\mathbf{R}^2$ (ordered pairs of real numbers).

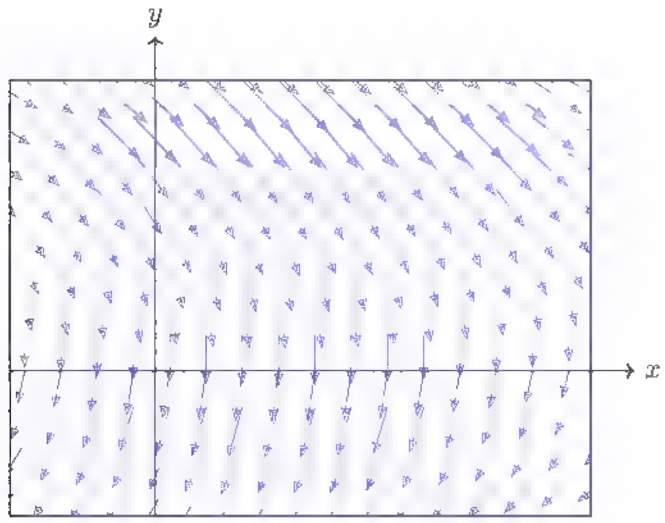


FIGURE A.1.1. Wind in some region R, as a function $\mathbf{w}$ over the region.

Figure A.1.1 is a visual representation of the meteorologist's data. The direction of the arrow at $\mathbf{x}$ is determined by the first component of $\mathbf{w}(\mathbf{x})$, and the *length* of each arrow is the speed of the wind at $\mathbf{x}$ (i.e., the second component of $\mathbf{w}(\mathbf{x})$). Strictly speaking, it is more natural to think about the output of the wind function in terms of vectors as in the main text (see Example 8.1.9). ■

There are many situations that require the use of functions with multiple numerical inputs (as in Example A.1.3) and/or multiple numerical outputs (as in Example A.1.4). Such functions are called *multivariable functions*; functions in your previous study of calculus are not of this type.

A.2. Functions in general, domain, target. At the most basic level, here is how general functions work. For sets X and Y, a *function* f from X to Y is a rule that assigns to each element x in X a single element, called $f(x)$, in Y. Figure A.2.1 below is a way to visualize what goes on with a function f from X to Y: to find $f(x)$, follow the arrow from the dot labeled x in X to the dot labeled $f(x)$ in Y.

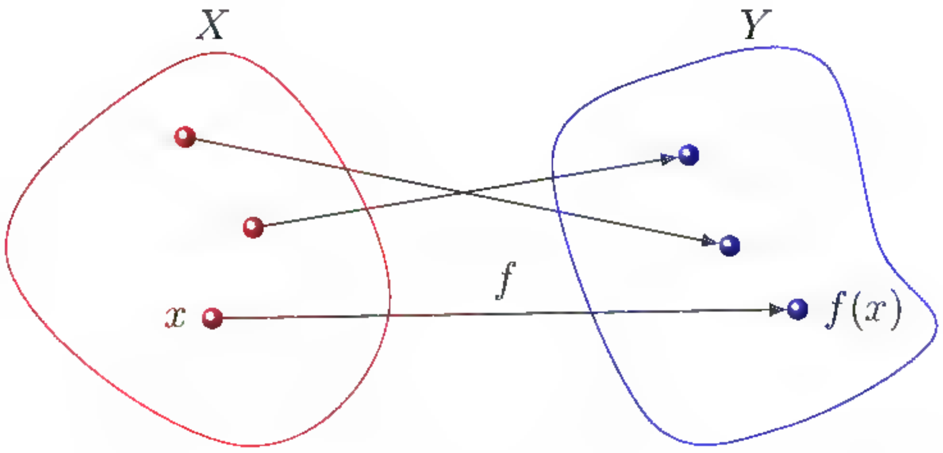


FIGURE A.2.1. Cartoon illustration of a typical function, from a set X to a set Y

To indicate that f is a function from X to Y, we write

$$f : X \to Y$$

(see Table 0.0.1). You might think of f as a device or a computer program that takes *inputs* from the set X (called the *domain* of f) and returns *outputs* in the set Y (the *target* of f).

When you first learned about functions, you probably saw examples like Example A.1.1 that define a function f in terms of an algebraic rule or formula it uses to assign outputs to inputs. For $f(x) = 3x/(x-1)$ as in that example, the picture as in Figure A.2.1 could have some explicit numbers, such as:

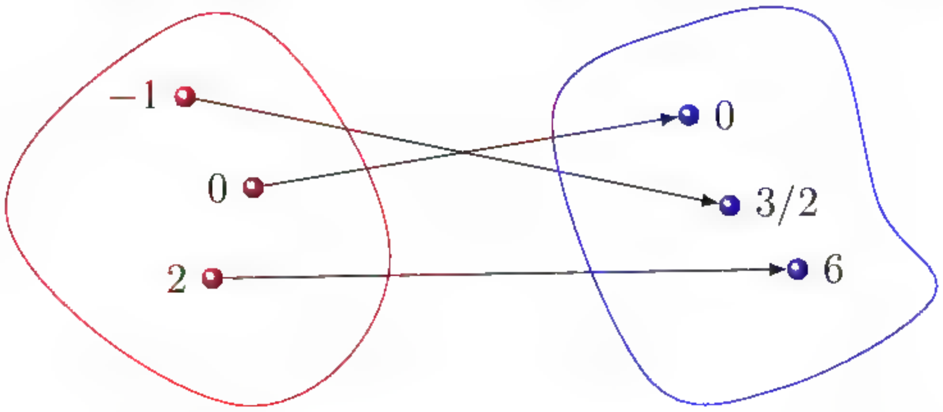


FIGURE A.2.2. Numerical data for the specific function $f(x) = 3x/(x-1)$

The reason there is an arrow in Figure A.2.2 pointing from the label -1 to the label $3/2$ is that we plug $x = -1$ into the rule and obtain $f(-1) = 3/2$. The method of defining a function in terms of a rule is convenient but it does not always make clear what the function's domain and target are.

Convention. If a single-variable function f is defined by a formula and no other indication is given, assume the domain X is the set of all real numbers x for which the formula $f(x)$ makes sense, and that the target Y is $\mathbf{R}$. This choice of domain is sometimes called the *largest possible domain* for f.

Example A.2.1. Here are some more functions:

$$f(x) = x^2 - 3x - 1 \quad g(x) = \tan x \quad h(x) = \ln\left|\frac{x-1}{x+1}\right| \quad s(x) = \sqrt{1-x^2}$$

These formulas define functions as *rules* for assigning outputs to inputs.

Their largest possible domains work out as follows: for f it is $\mathbf{R}$, for g it is all x not of the form $(k + 1/2)\pi$ for integers k (we need $\cos(x) \neq 0$), for h is it all $x \neq \pm 1$ (we need the denominator $x + 1$ to be nonzero, which is to say $x \neq -1$; also the non-negative $|(x-1)/(x+1)|$ must be nonzero because $\ln(t)$ only makes sense when $t > 0$, forcing $x \neq 1$ from the numerator), and for s it is $-1 \leq x \leq 1$ (we need $1 - x^2 \geq 0$ so that the square root make sense, and that says $|x| \leq 1$). ■

There are contexts in which we may be more judicious in defining the domain and target for a function:

(i) In Example A.1.2, the population function was given as $P(t) = P(0) \cdot 2^{t/13.4}$. Even though the function $P(t)$ as written makes sense for negative values of t, it may not make practical sense to include these in the *domain* of P because prior to the beginning of the experiment there was no population. (However, if the "initial time" refers to when the biologist began to make her measurements and the bacteria were present earlier then "negative time" could refer to time prior to the measurements.) Likewise, we may want to restrict the target to consist only of non-negative numbers since the population cannot be negative. In general, the way a mathematical model is being applied may determine our judgment for what to consider as a reasonable domain and target for a function.

(ii) For the gravitational force function F in Example A.1.3, a reasonable domain consists of those (m_1, m_2, r) in $\mathbf{R}^3$ whose coordinates are all > 0. Of course, we can work mathematically with the formula for F allowing $m_1, m_2 \leq 0$ too and also $r < 0$, though that wouldn't have any *physical* meaning in terms of the context of the mathematical model. However, the *same* formula shows up in Coulomb's Law for electrical forces, where the role of masses m_1, m_2 is replaced by charges q_1, q_2:

$$F = k_e \frac{q_1 q_2}{r^2}.$$

(here k_e is *Coulomb's constant*). In this latter context it *does* make physical sense to have q_1, q_2 of either sign (charge occurs in both positive and negative senses, unlike mass). Likewise the electrical force can be either positive or negative (since electrical forces can be both attractive or repulsive, whereas gravity is only attractive).

Whether for gravity or electrical forces, the *mathematical formula* is the same (even though the physical meaning of the variables is not). So it is useful to distinguish between the mathematics of a given formula and the context of a specific application to determine an appropriate domain (and target).

(iii) The target of the wind function in Example A.1.4 is a subset of $\mathbf{R}^2$. Because speed is ≥ 0, it is reasonable to view the second component of the output inside $[0, \infty)$. The first component (direction) is an angle. If the meteorologist represents angles in degrees, a reasonable choice for the target is the set of pairs (θ, s) with $0 \leq \theta < 360$ and $s \geq 0$.

A.3. Function facts, image, level sets. Here is a summary of some general facts:

(i) Any function $X \to Y$ must give an output in Y for *every* input x in X. Since $f(x) = 3x/(x-1)$ makes no sense when $x = 1$, for this function we must exclude $x = 1$ from the domain because there is no corresponding output in the target (which is $\mathbf{R}$), as indicated in Figure A.3.1.

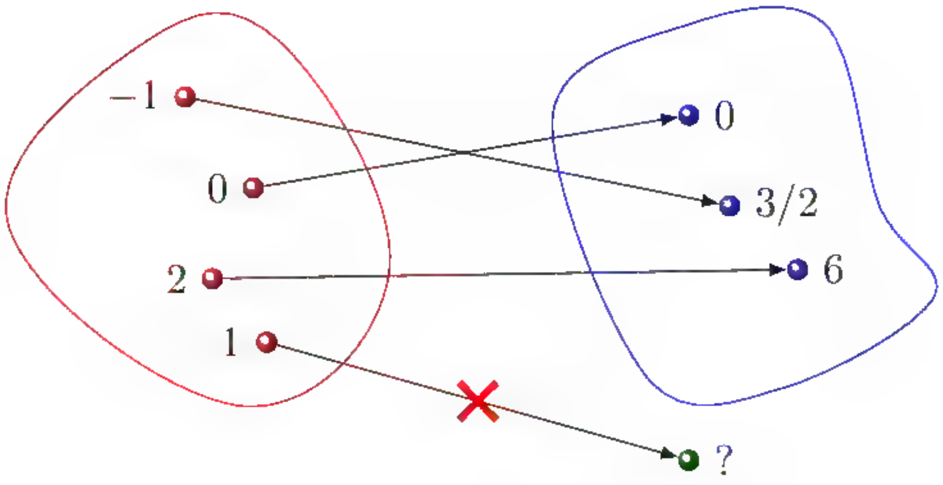


FIGURE A.3.1. The domain excludes points where the function makes no sense.

The largest possible domain for this example is the set S of all real numbers *except* 1. Since $f(x)$ is defined for all points x in S, we have a function $f : S \to \mathbf{R}$.

(ii) For a function f with a given domain X, it is necessary that every input x from X yield **only one** output $f(x)$ in Y. A situation as in Figure A.3.2 below does not depict a function because there is a point on the left that has two corresponding "outputs" on the right. This is forbidden!

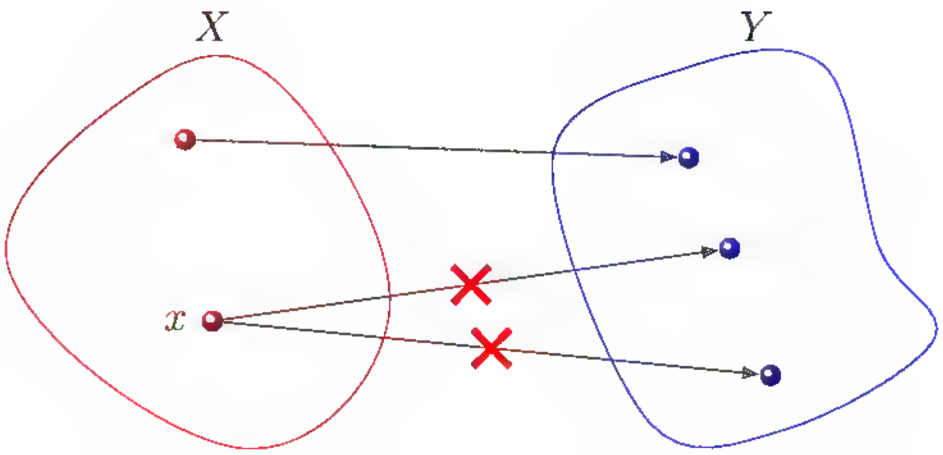


FIGURE A.3.2. A "rule" that is not a function

You may have learned about the "vertical line test" for single-variable functions $I \to \mathbf{R}$ on an interval I in the real number line. This expresses exactly the same requirement: if $f : X \to \mathbf{R}$ is an $\mathbf{R}$-valued function on some some subset X of $\mathbf{R}$ then any vertical line (i.e., a line parallel to the y-axis) may intersect the graph of $y = f(x)$ at most once (possibly not at all; e.g., the function $f(x) = \sqrt{x}$ on the domain of values $x \geq 0$ does not touch the vertical line $x = -2$). The curve in Figure A.3.3 is *not* the graph of a function because the dotted vertical line $x = 3$ passes through it at two points:

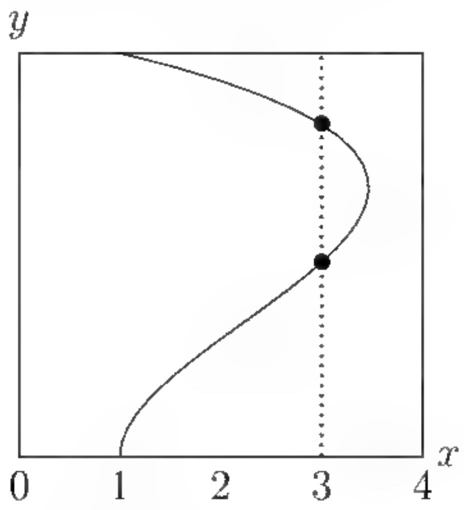


FIGURE A.3.3. Failure of the vertical line test

(iii) On the other hand, it is certainly fine for two different inputs to a function to result in the same output: there may be x_1, x_2 in X for which $x_1 \neq x_2$ but $f(x_1) = f(x_2)$. This is illustrated in Figure A.3.4.

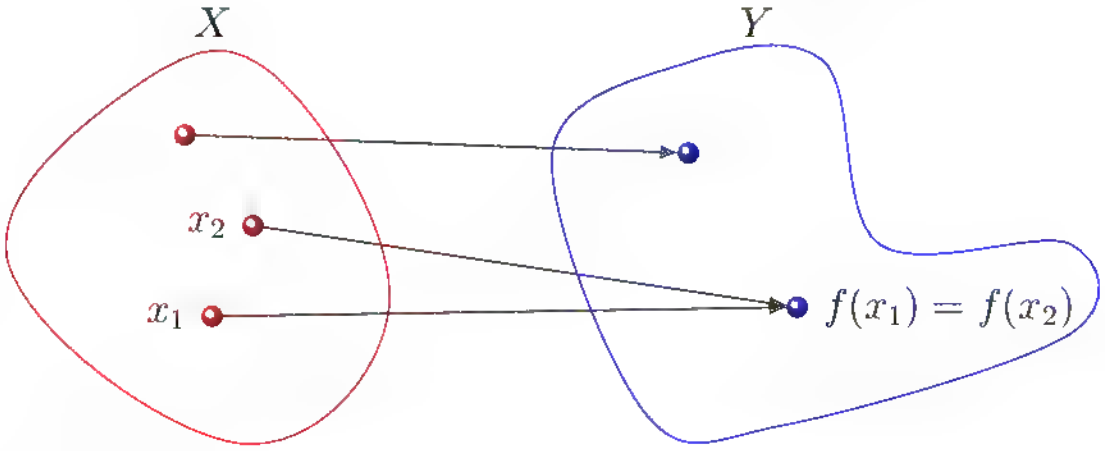


FIGURE A.3.4. A function f that carries two different inputs to the same output

As an example, consider $f(x) = x^2$, whose graph is shown in Figure A.3.5.

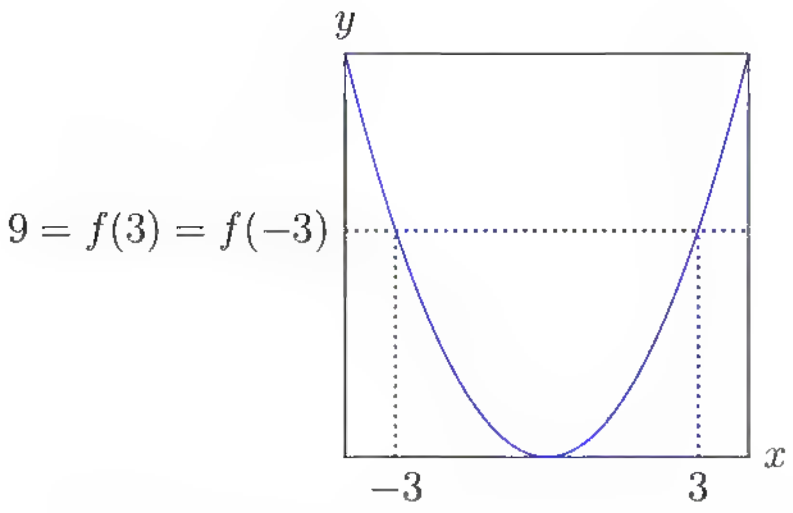


FIGURE A.3.5. There is no "horizontal line test" for functions!

We have $-3 \neq 3$, but $(-3)^2 = 9 = 3^2$. Graphically, this means that the parabola $y = x^2$ intersects the horizontal line $y = 9$ above $x = -3$ and above $x = 3$.

(iv) Finally, it is not necessary that *every* y in Y is an output of f. Using notation from Table 0.0.1, the set

$$\{y \in Y : y = f(x) \text{ for some } x \in X\}$$

is called the *image* of f (and is sometimes denoted $f(X)$). It is a *subset* of Y, but it doesn't have to be *equal* to the entire target Y. In the cartoon version in Figure A.3.6, the image of a function $f : X \to Y$ is expressed as the region inside the dashed outline. The target Y contains a marked point y that lies outside the image of f.

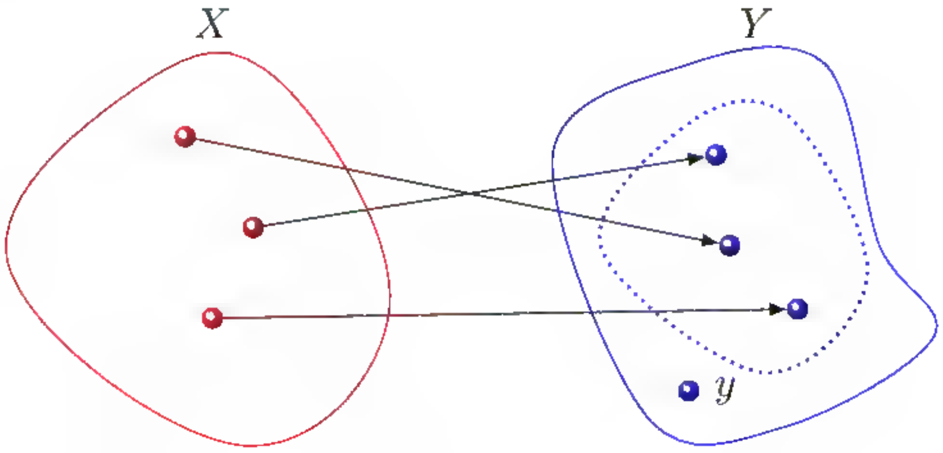


FIGURE A.3.6. The image of a function $f : X \to Y$ might not fill up Y.

Example A.3.1. In the case of the function $f(x) = 3x/(x-1)$ from Example A.2.1 that has domain consisting of all real numbers $x \neq 1$ and has target $\mathbf{R}$, the image consists of all y for which the equation $3x/(x-1) = y$ has a solution.

Carrying out the algebra to "solve for x in terms of y", we have

$$3x = (x-1)y = xy - y,$$

so $x(3-y) = -y$ and hence

$$x = \frac{-y}{3-y} = \frac{y}{y-3}$$

as long as we avoid division by 0, and all steps are reversible. Thus, as long as $y \neq 3$ we can always find a suitable x. On the other hand, when $y = 3$ there is no possible x since the equation $x(3-y) = -y$ with $y = 3$ becomes $x \cdot 0 = -3$ that is impossible. Hence, the image consists of all real numbers $y \neq 3$.

Another way to see what is going on in this example is to observe that

$$\frac{3x}{x-1} = \frac{3((x-1)+1)}{x-1} = 3 + \frac{1}{x-1},$$

so its graph is a hyperbola with vertical asymptote at $x = 1$ and shifted up 3 units. Hence, the horizontal asymptote on the x-axis for the hyperbola graph of $y = 1/(x-1)$ shifts up 3 units, corresponding to the omission of $y = 3$ from the image. Of course, this is all rather specific to the function $3x/(x-1)$ and one *cannot* expect the determination of the image of very complicated functions to be feasible; this is usually an extremely difficult or hopeless (and fortunately often totally unnecessary) task. ■

Example A.3.2. For another example that distinguishes between the target and the image, consider again $f(x) = x^2$ as a function $\mathbf{R} \to \mathbf{R}$. The target here is given as $\mathbf{R}$ (which makes sense because the square of a real number is always a real number), but the image of f is $[0, \infty)$, so they are not equal (because the target contains negative numbers whereas the image does not). This is fine!

For a more complicated function, such as $f(x) = 3x^4 - 5x^3 + 7x - 10$, it may be hard to explicitly describe the image. This is why it is often convenient to work with a target that is "big enough" (such as $\mathbf{R}$ in single-variable calculus) and not worry too much about determining the image exactly. ■

Example A.3.3. A real-world example where the target and image do not coincide can be found in Example A.1.4: even though the second component (wind speed) takes values in the unbounded interval $[0, \infty)$, there (hopefully) aren't any points at which the wind is faster than 300 mph, say. That is, even though the target of $\mathbf{w}$,

$$\{(\theta, s) : 0 \leq \theta < 360, s \geq 0\}$$

contains the point $(30, 1000000)$, presumably this point lies *outside the image* of $\mathbf{w}$ because its second coordinate is unreasonably large. And since it would be artificial to specify a definite upper bound on any possible wind speed, it is convenient to simply allow all $s \geq 0$ as second components in the target for the purposes of working with the mathematical model. ■

If $f : X \to Y$ is a function and y belongs to Y, the *level set of f at level y* is the set of all x in X for which $f(x) = y$. Once again using notation from Table 0.0.1, this is denoted as follows:

$$\{x \in X : f(x) = y\}.$$

In the cartoon version shown in Figure A.3.7, the level set at level y is the region inside the dashed outline.

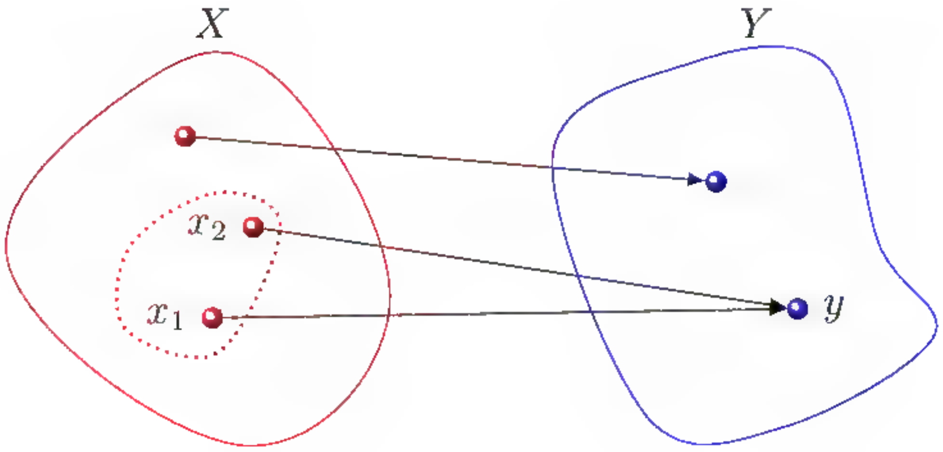


FIGURE A.3.7. Illustration of level set at level y, for a point y in Y.

Example A.3.4. For the function $f : \mathbf{R} \to \mathbf{R}$ given by squaring (i.e., $f(x) = x^2$), the level set of f at level 3 is the set $\{\sqrt{3}, -\sqrt{3}\}$. ■

It can happen that a level set is *empty*! Namely, the level set of f at some level y is empty precisely when the equation $f(x) = y$ has no solution x in X. Consideration of this possibility is important in more advanced mathematics and arises in the following example.

Example A.3.5. For $f(x) = x^2$ as a function $\mathbf{R} \to \mathbf{R}$, the level set of f at level $y = 0$ is $\{0\}$, at $y = 4$ it is $\{-2, 2\}$, and at $y = -5$ it is empty. Note that if instead we consider f as a function $[0, \infty) \to \mathbf{R}$ then the level set at level $y = 4$ shrinks to just be $\{2\}$.

A nice way to visualize the level sets in this case is to graph the parabola $y = f(x)$ across the domain of f (such as $X = \mathbf{R}$ or $X = [0, \infty)$, the latter corresponding to the right half of the parabola graph) and find the x-coordinates of the points where the graph meets the horizontal lines $y = 0$, $y = 4$, and $y = -5$. ■

"People don't understand how I can visualize four or five dimensions. Five-dimensional shapes are hard to visualize, but it doesn't mean you can't think about them. Thinking is really the same as seeing."

W. Thurston, 1982 Fields Medalist

"In the higher dimensions you cannot see everything, so you must have something, some tool, to guess or formulate things. And the tool was algebra, unquestionably algebra."

H. Hironaka, 1970 Fields Medalist

B. Further details on linear algebra results (optional)

In this appendix, we provide some proofs omitted from the main text. Using ideas related to the Gram–Schmidt process, in Section B.1 we prove certain properties of bases and dimension upon which our development depends: Theorem 4.2.8, Theorem 5.2.2, and Theorem 19.2.3, as well as the Gram–Schmidt process itself. Then in Section B.2 we show that two perspectives on the concept of "subspace" coincide, thereby unifying the geometric concept of span and the algebraic concept of null space. The final two sections are of a more advanced nature: in Section B.3 we give a proof of the Spectral Theorem (Theorem 24.1.4), and in Section B.4 we use the Spectral Theorem (or rather its consequence (24.2.2)) to give a geometric proof of the singular value decomposition (Theorem 27.3.3), complementing the matrix-algebra proof sketched in Section 27.4.

B.1. Dimension. The first step in the development of dimension in linear algebra is to show that the span V of a collection of k vectors in $\mathbf{R}^n$ has an *orthogonal* spanning set consisting of at most k nonzero vectors:

Proposition B.1.1. Let $\mathbf{v}_1, \dots, \mathbf{v}_k$ be a collection of vectors spanning a nonzero linear subspace V of $\mathbf{R}^n$. Then V is spanned by an *orthogonal* collection of at most k nonzero vectors.

PROOF. It is harmless to drop any $\mathbf{v}_i$'s that are equal to $\mathbf{0}$ (that only makes k become smaller without affecting the span V), so we may and do assume now that all $\mathbf{v}_i$'s are nonzero. If $k = 1$ then there is nothing to do: $\mathbf{v}_1$ is an orthogonal spanning set for V. Now we will work our way up, by increasing k one step at a time. This type of proof is called "mathematical induction."

Let's consider $k > 1$ and *suppose* we have already established the result we want for *all* collections of *fewer* than k nonzero vectors; that is, suppose that for any $h < k$ and vectors $\mathbf{w}_1, \dots, \mathbf{w}_h$ in $\mathbf{R}^n$, their span is also spanned by an orthogonal collection of at most h nonzero vectors. Note that this supposition is satisfied for $k = 2$ since the only $h < 2$ is $h = 1$ and we have treated the general case of a single nonzero vector. Such a supposition is called an "inductive hypothesis". We claim that under such a hypothesis, the desired result holds for any collection of k nonzero vectors. Once this is shown, we would obtain our desired result for $k = 2$ (since the inductive hypothesis does hold for $k = 2$). But with the desired result for $k = 2$ thereby established in general, we obtain the validity of the inductive hypothesis for $k = 3$, so we could run the same reasoning to bootstrap to establish the desired result for $k = 3$. That in turn establishes the validity of the inductive hypothesis for $k = 4$, and so on to handle all possible values of k.

If any among $\mathbf{v}_2, \dots, \mathbf{v}_k$ are scalar multiples of $\mathbf{v}_1$ then such a $\mathbf{v}_j$ can be dropped from the collection without affecting the span V, so V would be the span of $k - 1$ vectors. Since $k - 1 < k$, the inductive hypothesis would then ensure that V is the span of an orthogonal collection of at most $k - 1$ nonzero vectors; this is also a collection of at most k vectors, so we would be done. Hence, we can assume that each of $\mathbf{v}_2, \dots, \mathbf{v}_k$ is *not* a scalar multiple of $\mathbf{v}_1$, so by Theorem 7.1.1 applied to the pair of vectors $\mathbf{v}_1$ and $\mathbf{v}_j$ for each $2 \leq j \leq k$ we have

$$\operatorname{span}(\mathbf{v}_j, \mathbf{v}_1) = \operatorname{span}(\mathbf{v}'_j, \mathbf{v}_1)$$

with $\mathbf{v}'_j = \mathbf{v}_j - \mathbf{Proj}_{\mathbf{v}_1}(\mathbf{v}_j) \neq \mathbf{0}$ (non-vanishing since any projection $\mathbf{Proj}_{\mathbf{v}_1}(\mathbf{x})$ is a scalar multiple of $\mathbf{v}_1$, yet we have arranged that $\mathbf{v}_j$ is not a scalar multiple of $\mathbf{v}_1$). In particular,

$$\mathbf{v}_j = \mathbf{v}'_j + \mathbf{Proj}_{\mathbf{v}_1}(\mathbf{v}_j) = \mathbf{v}'_j + c_j\mathbf{v}_1$$

for some scalar c_j, so any linear combination $\sum_{i=1}^k a_i\mathbf{v}_i$ (i.e., anything in V) can be rewritten as

$$a_1\mathbf{v}_1 + \sum_{j=2}^k a_j\mathbf{v}_j = a_1\mathbf{v}_1 + \sum_{j=2}^k a_j(\mathbf{v}'_j + c_j\mathbf{v}_1) = (a_1 + \sum_{j=2}^k a_jc_j)\mathbf{v}_1 + \sum_{j=2}^k a_j\mathbf{v}'_j$$

(where the second equality involves collecting together all $\mathbf{v}_1$-terms).

In other words, everything in V is linear combination of $\mathbf{v}_1, \mathbf{v}'_2, \dots, \mathbf{v}'_k$. But by definition each $\mathbf{v}'_j$ is a linear combination of $\mathbf{v}_j$ and $\mathbf{v}_1$, so by a similar substitution calculation as above we obtain that $\mathrm{span}(\mathbf{v}_1, \mathbf{v}'_2, \dots, \mathbf{v}'_k) \subset V$. Since we already saw above that $V \subset \mathrm{span}(\mathbf{v}_1, \mathbf{v}'_2, \dots, \mathbf{v}'_k)$, we conclude that

$$V = \mathrm{span}(\mathbf{v}_1, \mathbf{v}'_2, \dots, \mathbf{v}'_k).$$

This says that V is also the span of the k nonzero vectors $\mathbf{v}_1, \mathbf{v}'_2, \dots, \mathbf{v}'_k$. What have we gained? The point is that the vectors $\mathbf{v}'_2, \dots, \mathbf{v}'_k$ are all orthogonal to $\mathbf{v}_1$ by design, so everything in their span

$$V' = \mathrm{span}(\mathbf{v}'_2, \dots, \mathbf{v}'_k)$$

is also orthogonal to $\mathbf{v}_1$. But V' is the span of $k-1$ nonzero vectors, so by the inductive hypothesis it is the span of an orthogonal collection of nonzero vectors $\mathbf{w}_1, \dots, \mathbf{w}_d$ for some $d \leq k-1$.

Since $\mathbf{v}_1$ is orthogonal to *everything* in V', and everything in V has the form $c_1\mathbf{v}_1 + \mathbf{v}'$ for some scalar c_1 and some $\mathbf{v}' \in V'$ (this is just separating off the $\mathbf{v}_1$-term in a linear combination of $\mathbf{v}_1, \mathbf{v}'_2, \dots, \mathbf{v}'_k$), it follows that $\mathrm{span}(\mathbf{v}_1, \mathbf{w}_1, \dots, \mathbf{w}_d) = V$. This spanning set for V consists of $1 + d \leq k$ nonzero vectors and it is a *pairwise orthogonal* collection (because everything in V' is orthogonal to $\mathbf{v}_1$). We have achieved the desired goal of building an orthogonal spanning set for V consisting of at most k nonzero vectors. □

Next, we prove a result on orthogonal complements that will be helpful in some subsequent arguments.

Proposition B.1.2. Suppose $\mathbf{v}_1, \dots, \mathbf{v}_k$ is an orthogonal collection of nonzero vectors in $\mathbf{R}^n$ spanning a linear subspace $V \subset \mathbf{R}^n$, with $k > 1$. For any nonzero $\mathbf{v} \in V$, the collection $\mathbf{v}^\perp$ of vectors in V that are perpendicular to $\mathbf{v}$ is a linear subspace, moreover spanned by $k-1$ vectors.

PROOF. We will adapt the calculation given in Example 4.1.6 to a more general setting. We have $\mathbf{v} = c_1\mathbf{v}_1 + \cdots + c_k\mathbf{v}_k$ for some scalar coefficients c_i at least one of which is nonzero since $\mathbf{v}$ is nonzero. Any vector $\mathbf{x} \in V$ can be written as $\mathbf{x} = x_1\mathbf{v}_1 + \cdots + x_k\mathbf{v}_k$ for scalars $x_1, \dots, x_k$. Since the $\mathbf{v}_i$'s are pairwise orthogonal, to show that $\mathbf{x}$ is perpendicular to $\mathbf{v}$ means showing the vanishing of

$$\mathbf{v} \cdot \mathbf{x} = \sum_{j=1}^k c_j(\mathbf{v}_j \cdot \mathbf{x}) = \sum_{j=1}^k c_j(\mathbf{v}_j \cdot \sum_{i=1}^k x_i\mathbf{v}_i) = \sum_{j=1}^k c_j(\sum_{i=1}^k \mathbf{v}_j \cdot (x_i\mathbf{v}_i)).$$

But $\mathbf{v}_j \cdot (x_i\mathbf{v}_i) = x_i(\mathbf{v}_j \cdot \mathbf{v}_i)$ vanishes if $i \neq j$ by orthogonality, so all terms in each inner sum vanish except possibly when $i = j$, leaving us with

$$\mathbf{v} \cdot \mathbf{x} = \sum_{j=1}^k c_jx_j(\mathbf{v}_j \cdot \mathbf{v}_j).$$

At least one of the coefficients $c_1, \dots, c_k$ is nonzero, say $c_i \neq 0$, so the vanishing of $\mathbf{v} \cdot \mathbf{x}$ is encoded by solving the equation

$$\sum_{j=1}^k c_jx_j(\mathbf{v}_j \cdot \mathbf{v}_j) = 0$$

for x_i in terms of the other x_j's as we did in Example 4.1.6: in the present generality it comes out as

$$x_i = \sum_{j \neq i} -\frac{c_j(\mathbf{v}_j \cdot \mathbf{v}_j)}{c_i(\mathbf{v}_i \cdot \mathbf{v}_i)} x_j.$$

Thus, separating the term $x_i\mathbf{v}_i$ from the rest in the expression $\mathbf{x} = \sum_{j=1}^{k} x_j\mathbf{v}_j$ and defining $c'_j = c_j(\mathbf{v}_j \cdot \mathbf{v}_j)$ for ease of notation yields

$$\mathbf{x} = x_i\mathbf{v}_i + \sum_{j \neq i} x_j\mathbf{v}_j = \sum_{j \neq i} -(c'_j/c'_i)x_j\mathbf{v}_i + \sum_{j \neq i} x_j\mathbf{v}_j = \sum_{j \neq i} x_j(-(c'_j/c'_i)\mathbf{v}_i + \mathbf{v}_j).$$

This shows that every $\mathbf{x}$ perpendicular to $\mathbf{v}$ is contained in the span of the $k-1$ vectors $\mathbf{w}_j = -(c'_j/c'_i)\mathbf{v}_i + \mathbf{v}_j$ for $j \neq i$. We claim that $\mathbf{v}^{\perp}$ coincides with the span W of those $k-1$ vectors (and so in particular $\mathbf{v}^{\perp}$ is a linear subspace of $\mathbf{R}^n$). Having shown that $\mathbf{v}^{\perp}$ is contained inside the span W of the $k-1$ vectors $\mathbf{w}_j = -(c'_j/c'_i)\mathbf{v}_i + \mathbf{v}_j$ for $j \neq i$, it remains to show that everything in W belongs to $\mathbf{v}^{\perp}$.

Provided that each $\mathbf{w}_j$ belongs to $\mathbf{v}^{\perp}$, so does everything in their span since any linear combination of vectors in $\mathbf{v}^{\perp}$ is also contained in $\mathbf{v}^{\perp}$. Indeed, if $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbf{v}^{\perp}$ then $c_1\mathbf{x}_1 + \cdots + c_m\mathbf{x}_m \in \mathbf{v}^{\perp}$ for any scalars $c_1, \dots, c_m$ since

$$\mathbf{v} \cdot (c_1\mathbf{x}_1 + \cdots + c_m\mathbf{x}_m) = c_1(\mathbf{v} \cdot \mathbf{x}_1) + \cdots + c_m(\mathbf{v} \cdot \mathbf{x}_m) = c_1(0) + \cdots + c_m(0) = 0.$$

Thus, we just have to check that $\mathbf{w}_j \in \mathbf{v}^{\perp}$ for each $j \neq i$, and this is a direct calculation: orthogonality of $\mathbf{v}_1, \dots, \mathbf{v}_k$ implies $\mathbf{v} \cdot \mathbf{v}_h = \sum_{r=1}^{k}(c_r(\mathbf{v}_r \cdot \mathbf{v}_h)) = c_h(\mathbf{v}_h \cdot \mathbf{v}_h)$, so

$$\mathbf{v} \cdot \mathbf{w}_j = \mathbf{v} \cdot \left(-\frac{c'_j}{c'_i}\mathbf{v}_i + \mathbf{v}_j\right) = -\frac{c'_j}{c'_i}(\mathbf{v} \cdot \mathbf{v}_i) + \mathbf{v} \cdot \mathbf{v}_j = -\frac{c'_j}{c'_i}(c_i(\mathbf{v}_i \cdot \mathbf{v}_i)) + c_j(\mathbf{v}_j \cdot \mathbf{v}_j).$$

But $c'_i = c_i(\mathbf{v}_i \cdot \mathbf{v}_i)$ and $c'_j = c_j(\mathbf{v}_j \cdot \mathbf{v}_j)$ by definition, so we have shown $\mathbf{v} \cdot \mathbf{w}_j = -(c'_j/c'_i)c'_i + c'_j = -c'_j + c'_j = 0$ as desired. ☐

Next, we show that for any nonzero linear subspace $W \subset \mathbf{R}^n$, any two orthogonal spanning sets of nonzero vectors for W have the same size. That is:

Proposition B.1.3. Let $W \subset \mathbf{R}^n$ be a nonzero subspace. Any two sets of nonzero orthogonal vectors that span W have the same size.

PROOF. Letting m be the size of one such spanning set for W, we want to show that all such for W have size m. First suppose $m = 1$. Then $W = \text{span}(\mathbf{w})$ is the span of a single vector, and so any two nonzero vectors in W have the form $c\mathbf{w}$ and $c'\mathbf{w}$ for some *nonzero* scalars c, c'. Such vectors cannot be orthogonal to each other: $(c\mathbf{w}) \cdot (c'\mathbf{w}) = (cc')(\mathbf{w} \cdot \mathbf{w}) = (cc')\|\mathbf{w}\|^2$ is nonzero since $\mathbf{w} \neq \mathbf{0}$ and $c, c' \neq 0$. Hence, no *two* nonzero vectors $\mathbf{w}_1, \mathbf{w}_2 \in W$ are ever orthogonal to one another, so when $m = 1$ a collection of orthogonal nonzero vectors spanning W must consist of exactly one vector.

Now we will work our way up, by increasing m one step at a time. Let's consider $m > 1$ and make the "inductive hypothesis" that we have already established the statement of Proposition B.1.3 for any nonzero subspace of $\mathbf{R}^n$ that is the span of an orthogonal collection of *fewer* than m nonzero vectors (e.g., this is satisfied for $m = 2$ since we have treated the case of the span of a single nonzero vector). We claim that under such a hypothesis, the desired result holds for the span of any orthogonal collection of m nonzero vectors. Once this is shown then we would obtain our desired result for $m = 2$ (since the inductive hypothesis does hold for $m = 2$), but the established result for $m = 2$ thereby establishes the validity of the inductive hypothesis for $m = 3$, so we could run the same reasoning to obtain the desired result for $m = 3$. That in turn establishes the validity of the inductive hypothesis for $m = 4$, and so on to handle all possibilities.

Suppose that for *any* span of a (non-empty) orthogonal collection of fewer than m nonzero vectors it is known that all spanning sets consisting of pairwise orthogonal nonzero vectors have the same size. We aim to prove the same for the span W of m pairwise orthogonal nonzero vectors $\mathbf{w}_1, \ldots, \mathbf{w}_m$. In other words, if $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ is an orthogonal collection of nonzero vectors spanning W then we claim that $k = m$. The possibility $k < m$ cannot happen. Indeed, if $k < m$ then we could apply the inductive hypothesis to conclude that *all* spanning sets of W consisting of pairwise orthogonal nonzero vectors have size k. This contradicts the fact that $\{\mathbf{w}_1, \ldots, \mathbf{w}_m\}$ is a spanning set for W containing m nonzero orthogonal vectors with $m > k$. So indeed we cannot have $k < m$, so necessarily $k \geq m$.

The key idea to rule out the possibility $k > m$ is to consider *the collection* $W' = \mathbf{x}_1^\perp$ *of the vectors in* W *orthogonal to* $\mathbf{x}_1$. On the one hand, since W is the span of the collection $\mathbf{w}_1, \ldots, \mathbf{w}_m$ of m pairwise orthogonal nonzero vectors, by Proposition B.1.2 we know that W' is spanned by $m - 1$ vectors. By Proposition B.1.1, it follows that W' has an orthogonal spanning set of nonzero vectors consisting of *at most* $m - 1$ vectors. If the size of the spanning set thereby obtained is denoted $d \leq m - 1$ then by the "inductive hypothesis" *all* spanning sets of pairwise orthogonal nonzero vectors for W' consist of exactly d vectors.

On the other hand, we claim that W' is the span of the orthogonal collection of $k - 1$ nonzero vectors $\mathbf{x}_2, \ldots, \mathbf{x}_k$. Indeed, a general vector $\mathbf{w} \in W$ has the form

$$\mathbf{w} = c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \cdots + c_k\mathbf{x}_k,$$

so

$$\mathbf{w} \cdot \mathbf{x}_1 = c_1(\mathbf{x}_1 \cdot \mathbf{x}_1) + c_2(\mathbf{x}_2 \cdot \mathbf{x}_1) + \cdots + c_k(\mathbf{x}_k \cdot \mathbf{x}_1) = c_1(\mathbf{x}_1 \cdot \mathbf{x}_1)$$

since $\mathbf{x}_j \cdot \mathbf{x}_1 = 0$ for all $j \neq 1$ (as the collection of $\mathbf{x}_i$'s is orthogonal). But $\mathbf{x}_1 \cdot \mathbf{x}_1 = \|\mathbf{x}_1\|^2$ is nonzero since $\mathbf{x}_1 \neq \mathbf{0}$, so $\mathbf{w} \cdot \mathbf{x}_1 = 0$ precisely when $c_1 = 0$, which is to say that $\mathbf{w}$ belongs to the span of $\{\mathbf{x}_2, \ldots, \mathbf{x}_k\}$. So indeed W' is the span of an orthogonal collection of $k - 1$ nonzero vectors. But we have already seen that *every* orthogonal collection of nonzero vectors spanning W' consists of $d \leq m - 1$ vectors, so $k - 1 = d \leq m - 1$ and hence $k \leq m$. It was already shown that $k \geq m$, so we conclude that $k = m$ as desired. □

We have shown in Propositions B.1.1 and B.1.3 that for each nonzero linear subspace V of $\mathbf{R}^n$, there is a spanning set of nonzero orthogonal vectors and every spanning set of nonzero orthogonal vectors for V has the same number of vectors. We call this common number the *ortho-dimension* of V and denote it as$\mathrm{odim}(V)$. We have $\dim(V) \leq \mathrm{odim}(V)$ since (i) V has a spanning containing $\mathrm{odim}(V)$ nonzero (orthogonal) vectors and (ii) $\dim(V)$ is (by definition) the smallest number of nonzero vectors in any possible spanning set for V. Now we can prove Theorem 5.2.2.

PROOF OF THEOREM 5.2.2. Our task is to prove $\mathrm{odim}(V) = \dim(V)$ for all nonzero subspaces $V \subset \mathbf{R}^n$. Since V is spanned by $\dim(V)$ vectors, Proposition B.1.1 provides a spanning set consisting of at most $\dim(V)$ pairwise orthogonal nonzero vectors! Any such spanning set has size $\mathrm{odim}(V)$ by Proposition B.1.3, so we get the reverse inequality $\mathrm{odim}(V) \leq \dim(V)$, forcing $\mathrm{odim}(V) = \dim(V)$. This completes the proof. □

With the equality of dimension and ortho-dimension now proved, we finally know that "orthogonal basis" for a nonzero subspace $V \subset \mathbf{R}^n$ is the same thing is "spanning set of pairwise orthogonal nonzero vectors". Finally, we can prove Theorem 4.2.8.

PROOF OF THEOREM 4.2.8. Let W, V be linear subspaces of $\mathbf{R}^n$ with $W \subset V$. We want to show that $\dim(W) \leq \dim(V)$ with equality precisely when $W = V$. If $W = \{\mathbf{0}\}$ (so $\dim(W) = 0$) then everything is clear, so we may assume W is nonzero (so also V is nonzero). By the equality of dimension and ortho-dimension, we may pick an orthogonal basis $\{\mathbf{w}_1, \dots, \mathbf{w}_m\}$ of W, with $m = \dim(W)$. If this extends to an orthogonal basis $\{\mathbf{w}_1, \dots, \mathbf{w}_m, \mathbf{v}_1, \dots, \mathbf{v}_r\}$ of V, then

$$\dim(V) = \operatorname{odim}(V) = m + r \geq m = \operatorname{odim}(W) = \dim(W)$$

and equality is exactly the case $r = 0$, which is to say that the $\mathbf{w}_i$'s constitute an orthogonal basis of V. But the span of the $\mathbf{w}_i$'s is equal to W by design, so they form an orthogonal basis of V precisely when $W = V$. Hence, it remains to show that any orthogonal collection of nonzero vectors $\mathbf{v}_1, \dots, \mathbf{v}_m$ in a linear subspace $V \subset \mathbf{R}^n$ may always be enlarged to an orthogonal basis $\mathbf{v}_1, \dots, \mathbf{v}_m, \dots, \mathbf{v}_d$ of V. This will be an argument via mathematical induction, where we shall now work up one step at a time based on the dimension of V.

If $\dim(V) = 1$ then V is the span of a single nonzero vector, so (as we saw near the start of the proof of Proposition B.1.3) there aren't even two orthogonal nonzero vectors to be found in V. This forces $m = 1$ in such cases, so there is nothing to do. Now suppose $d = \dim(V) > 1$ and that the desired result (to extend any collection of pairwise orthogonal nonzero vectors in a linear subspace to an orthogonal basis of the subspace) holds for all linear subspaces of $\mathbf{R}^n$ with dimension $< d$. We shall show the same holds for the d-dimensional V.

If $\mathbf{v}_1, \dots, \mathbf{v}_m$ spans V then this collection of vectors is an orthogonal basis of V and there is nothing to do. So now suppose its span does not exhaust V, and pick $\mathbf{v} \in V$ outside its span. (In particular, $\mathbf{v}$ is not a scalar multiple of any $\mathbf{v}_j$.) Define $\mathbf{v}' = \mathbf{v} - \mathbf{Proj}_V(\mathbf{v})$. This is nonzero (since $\mathbf{v} \notin V$) and belongs to $V^{\perp}$. Hence, $\mathbf{v}'$ is *orthogonal* to every $\mathbf{v}_i$.

Now apply Proposition B.1.2 to the nonzero vector $\mathbf{v}' \in V$, noting since V has an orthogonal basis of size d (as $\operatorname{odim}(V) = \dim(V)$) it follows that the linear subspace $\mathbf{v}'^{\perp} \subset V$ containing all $\mathbf{v}_i$'s is spanned by at most $d - 1$ vectors. Then Proposition B.1.1 tells us that $\dim(\mathbf{v}'^{\perp}) = \operatorname{odim}(\mathbf{v}'^{\perp}) \leq d - 1$, so the inductive hypothesis is applicable to the linear subspace $\mathbf{v}'^{\perp}$! Hence, the collection $\{\mathbf{v}_1, \dots, \mathbf{v}_m\}$ inside $\mathbf{v}'^{\perp}$ extends to an orthogonal basis of $\mathbf{v}'^{\perp}$. But Theorem 7.1.1 tells us that every vector $\mathbf{x} \in V$ that is not a scalar multiple of $\mathbf{v}'$ is the sum of the scalar multiple $\mathbf{Proj}_{\mathbf{v}'}(\mathbf{x})$ of $\mathbf{v}'$ and a vector belonging to $\mathbf{v}'^{\perp}$, so any spanning set of $\mathbf{v}'^{\perp}$ combined with $\mathbf{v}'$ is a spanning set for V! In particular, if $\{\mathbf{v}_1, \dots, \mathbf{v}_m, \dots, \mathbf{v}_k\}$ is an orthogonal basis of $\mathbf{v}'^{\perp}$ extending $\{\mathbf{v}_1, \dots, \mathbf{v}_m\}$ then

$$B = \{\mathbf{v}_1, \dots, \mathbf{v}_m, \dots, \mathbf{v}_k, \mathbf{v}'\}$$

is a spanning set for V consisting of nonzero vectors. But $\mathbf{v}'$ is orthogonal to everything else in this collection (as each of $\mathbf{v}_1, \dots, \mathbf{v}_k$ belongs to $\mathbf{v}'^{\perp}$), and the collection $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ is pairwise orthogonal by design. Hence, B is a collection of pairwise orthogonal nonzero vectors that spans V, so it is an orthogonal basis. This is an orthogonal basis of V that contains the initial orthogonal collection $\{\mathbf{v}_1, \dots, \mathbf{v}_m\}$, and so completes the inductive argument and so also the proof of Theorem 4.2.8. ☐

Before we can prove Theorem 19.2.3, whose formulation involves the Gram–Schmidt process, we need to show that the Gram–Schmidt process actually works. So let us now carry out that argument. The first step of the Gram–Schmidt process works as claimed, due to the definitions. There are k steps overall, and to check that everything works as claimed we proceed from each step to the next in an "inductive" manner. That is, for an integer j satisfying $1 \leq j < k$ we suppose everything has worked up through the end of the jth step, so we have an orthogonal basis $\mathcal{B}_j$ of $V_j = \operatorname{span}(\mathbf{v}_1, \dots, \mathbf{v}_j)$ consisting of the nonzero $\mathbf{w}_i$'s obtained so far. We want to show that everything works at the end of the next step. For that purpose, we define

$$\mathbf{w}_{j+1} = \mathbf{v}_{j+1} - \mathbf{Proj}_{V_j}(\mathbf{v}_{j+1})$$

and need to show two things: if $\mathbf{w}_{j+1} = \mathbf{0}$ then $V_{j+1} = V_j$, and if $\mathbf{w}_{j+1} \neq \mathbf{0}$ then combining $\mathcal{B}_j$ and $\mathbf{w}_{j+1}$ gives an orthogonal basis of V_{j+1}. (That would allow us to carry out the inductive argument, and upon reaching the end of the kth step we would have an orthogonal basis $\mathcal{B}_k$ of $V_k = V$ as claimed.)

From the definitions of V_j and V_{j+1} as spans, we see via the equality

$$\sum_{i=1}^{j+1} c_i \mathbf{v}_i = (\sum_{i=1}^{j} c_i \mathbf{v}_i) + c_{j+1}\mathbf{v}_{j+1}$$

that the vectors in V_{j+1} are precisely the sums $\mathbf{v}' + c\mathbf{v}_{j+1}$ for vectors $\mathbf{v}' \in V_j$ and scalars c. When $\mathbf{w}_{j+1} = \mathbf{0}$ we have $\mathbf{v}_{j+1} = \mathbf{Proj}_{V_j}(\mathbf{v}_{j+1}) \in V_j$, so for any $\mathbf{v}' \in V_j$ and scalar c certainly $\mathbf{v}' + c\mathbf{v}_{j+1} \in V_j$ (as a linear combination of vectors in a linear subspace belongs to that linear subspace). Hence, $V_{j+1} = V_j$ as desired when $\mathbf{w}_{j+1} = \mathbf{0}$.

Now suppose $\mathbf{w}_{j+1}$ is nonzero. By Theorem 6.2.1, the difference vector $\mathbf{w}_{j+1}$ between $\mathbf{v}_{j+1}$ and its projection into V_j is orthogonal to everything in V_j. But the expression $\mathbf{v}' + c\mathbf{v}_{j+1}$ for general elements of V_{j+1} can be rewritten as

$$\mathbf{v}' + c(\mathbf{w}_{j+1} + \mathbf{Proj}_{V_j}(\mathbf{v}_{j+1})) = (\mathbf{v}' + \mathbf{Proj}_{V_j}(\mathbf{v}_{j+1})) + c\mathbf{w}_{j+1},$$

which is the sum of a vector in V_j and a scalar multiple of $\mathbf{w}_{j+1}$. Since we have written everything in V_{j+1} as the sum of a vector in V_j and a scalar multiple of $\mathbf{w}_{j+1}$, and $\mathcal{B}_j$ is an orthogonal basis of V_j, it follows that $\mathcal{B}_j$ together with the nonzero vector $\mathbf{w}_{j+1}$ is a spanning set for V_{j+1} consisting of nonzero vectors. This collection is also orthogonal: we have seen that $\mathbf{w}_{j+1}$ is orthogonal to *everything* in V_j (hence to everything in $\mathcal{B}_j$), and by the inductive hypothesis $\mathcal{B}_j$ is a collection of pairwise orthogonal vectors. Hence, $\mathcal{B}_j$ together with $\mathbf{w}_{j+1}$ is an orthogonal basis of V_{j+1}. This completes the proof that the Gram–Schmidt process always works. Using this, we can prove Theorem 19.2.3:

PROOF OF THEOREM 19.2.3. By the design of the Gram–Schmidt process, the nonzero $\mathbf{w}_i$'s are exactly the members of the collection $\mathcal{B}_k$ that is an orthogonal basis of V, so the number of such $\mathbf{w}_i$'s is equal to $\mathrm{odim}(V) = \dim(V)$ as claimed at the start of Theorem 19.2.3.

Since there are k steps in the Gram–Schmidt process, to say that there are no nonzero $\mathbf{w}_i$'s is exactly to say that there are k nonzero $\mathbf{w}_i$'s. But the number of nonzero $\mathbf{w}_i$'s is known to always be equal to $\dim(V)$ (since we have shown above that the Gram–Schmidt process always works), so the equivalence of the conditions (i) and (ii) in Theorem 19.2.3 is established. When these equivalent conditions hold, so the spanning set $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ of nonzero vectors in V has minimal size, we have to show that this collection of vectors must be linearly independent. If $k = 1$ then there is nothing to do. Now suppose $k > 1$. We have to rule out the possibility that $\mathbf{v}_i$ is a linear combination of the rest. When the latter happens, $\mathbf{v}_i$ is redundant in the span: the collection of $\mathbf{v}_j$'s without $\mathbf{v}_i$ has the same span (since the contribution of $\mathbf{v}_i$ can always be absorbed into the span of the rest). But that is impossible: it would provide a collection of $k-1$ vectors whose span is equal to the linear subspace V, yet $\dim(V) = k$ (so all spanning sets of V have size at least k). We have shown that when the equivalent conditions (i) and (ii) in Theorem 19.2.3 hold, then (iii) also holds.

Finally, we have to show that when condition (iii) in Theorem 19.2.3 holds, then the equivalent conditions (i) and (ii) hold. It is enough to show that (ii) holds: all $\mathbf{w}_i$'s are nonzero. We suppose to the contrary that some $\mathbf{w}_i$ vanishes and seek a contradiction (so such a situation couldn't have arisen after all). Always $\mathbf{w}_1 = \mathbf{v}_1$ is nonzero, so necessarily $i \geq 2$. By definition,

$$\mathbf{w}_i = \mathbf{v}_i - \mathbf{Proj}_{V_{i-1}}(\mathbf{v}_i),$$

so the vanishing of some $\mathbf{w}_i$ forces

$$\mathbf{v}_i = \mathbf{Proj}_{V_{i-1}}(\mathbf{v}_i) \in V_{i-1} = \mathrm{span}(\mathbf{v}_1, \dots, \mathbf{v}_{i-1}).$$

This implies linear *dependence* for $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ (in fact, even linear dependence of the first i vectors in this collection), contradicting the assumption that (iii) holds. ☐

As an application of Theorem 19.2.3, we can finally establish Theorem 18.1.8 (which played no role in the proof of Theorem 19.2.3, so there is no circular reasoning involved).

Proposition B.1.4. Let A be an $m \times n$ matrix.

(i) If there exists an $n \times m$ matrix B for which $AB = \mathrm{I}_m$ and $BA = \mathrm{I}_n$ then $n = m$.
(ii) If $n = m$ and B is an $n \times n$ matrix for which $AB = \mathrm{I}_n$ then $BA = \mathrm{I}_n$ (so A is invertible with inverse B).

PROOF. By Theorem 19.2.3, a basis of any $\mathbf{R}^N$ is the same as a linearly independent spanning set (so in particular any linearly independent spanning set has the same size, which is the dimension). Hence, $\dim \mathbf{R}^n = n$ and $\dim \mathbf{R}^m = m$ (as can also be seen by the established equality of dimension and orthodimension), so to prove (i) it suffices to show that if such a B exists as there then $\mathbf{R}^n$ and $\mathbf{R}^m$ have the same dimension.

For the standard basis $\mathbf{e}_1, \dots, \mathbf{e}_n$ of $\mathbf{R}^n$, define $\mathbf{v}_j = A\mathbf{e}_j \in \mathbf{R}^m$. We shall prove that $\mathbf{v}_1, \dots, \mathbf{v}_n$ is a basis of $\mathbf{R}^m$, so $\dim \mathbf{R}^m = n$, forcing $m = n$ as desired. It is equivalent to show that $\mathbf{v}_1, \dots, \mathbf{v}_n$ is a linearly independent spanning set for $\mathbf{R}^m$.

We are assuming in (i) that there is an $n \times m$ matrix B for which $AB = \mathrm{I}_m$ and $BA = \mathrm{I}_n$. Every $\mathbf{v} \in \mathbf{R}^m$ has the form $A\mathbf{w}$ for some $\mathbf{w} \in \mathbf{R}^n$ since $\mathbf{v} = (AB)\mathbf{v} = A(B\mathbf{v})$ (so take $\mathbf{w}$ to be $B\mathbf{v}$). Writing $\mathbf{w} = \sum_{j=1}^n c_j \mathbf{e}_j$, we obtain

$$\mathbf{v} = A\mathbf{w} = \sum_{j=1}^n c_j A\mathbf{e}_j = \sum_{j=1}^n c_j \mathbf{v}_j,$$

so the $\mathbf{v}_j$'s span $\mathbf{R}^m$. By the same calculation, they must be linearly independent: if $\sum_{j-1}^n c_j \mathbf{v}_j = \mathbf{0}$ then

$$\mathbf{0} = \sum_{j=1}^n c_j \mathbf{v}_j = A(\sum_{j=1}^n c_j \mathbf{e}_j),$$

so multiplying on the left by B and using that $BA = \mathrm{I}_n$ yields that $\sum_{j-1}^n c_j \mathbf{e}_j$ vanishes. Such vanishing as an n-vector forces all c_j to vanish, as desired. This completes the proof of (i).

Turning to (ii), suppose A, B are $n \times n$ matrices for which $AB = \mathrm{I}_n$. We need to prove $BA = \mathrm{I}_n$. Letting $\mathbf{v}_j = B\mathbf{e}_j$, we claim that the vectors $\mathbf{v}_1, \dots, \mathbf{v}_n \in \mathbf{R}^n$ are linearly independent. Once this is shown, then this collection of n vectors has n-dimensional span (by Theorem 19.2.3), so its span coincides with $\mathbf{R}^n$ (by Theorem 4.2.8). In other words, if the $\mathbf{v}_j$'s are linearly independent then they constitute a *basis* of $\mathbf{R}^n$. This is the key fact from which (ii) will be deduced.

To show that the $\mathbf{v}_j$'s are linearly independent, suppose $c_1, \dots, c_n$ are scalars for which

$$\sum_{j=1}^n c_j \mathbf{v}_j = \mathbf{0}.$$

We want to show that the c_j's all vanish. Forming the matrix-vector product against A on both sides of this vector equality and using that $AB = \mathrm{I}_n$, we get

$$\mathbf{0} = A\mathbf{0} = A(\sum_{j=1}^n c_j \mathbf{v}_j) = \sum_{j=1}^n c_j A\mathbf{v}_j = \sum_{j=1}^n c_j A(B\mathbf{e}_j) = \sum_{j=1}^n c_j \mathbf{e}_j.$$

The equality of outer terms forces the c_j's to vanish, in view of how the $\mathbf{e}_j$'s are *defined*. This completes the proof that the $\mathbf{v}_j$'s are a basis of $\mathbf{R}^n$.

Now we are ready to prove that BA is the identity transformation. We will show that $(BA)\mathbf{v} = \mathbf{v}$ for all $\mathbf{v} \in \mathbf{R}^n$; applying this to $\mathbf{v} = \mathbf{e}_j$ would then imply that each column of BA agrees with the corresponding column of I_n, so we'd have $BA = \mathrm{I}_n$ as desired. Since the $\mathbf{v}_j$'s are a basis of $\mathbf{R}^n$, we can write $\mathbf{v} = \sum_{j=1}^n c_j \mathbf{v}_j$ for some scalars $c_1, \dots, c_n$. Hence,

$$(BA)\mathbf{v} = B(A\mathbf{v}) = B(\sum_{j=1}^n c_j A\mathbf{v}_j) = \sum_{j=1}^n c_j B(A\mathbf{v}_j).$$

We want this to be equal to $\mathbf{v} = \sum_{j=1}^n c_j \mathbf{v}_j$, so it is enough to show that $B(A\mathbf{v}_j) = \mathbf{v}_j$ for all j. By definition $\mathbf{v}_j = B(\mathbf{e}_j)$, and $AB = \mathrm{I}_n$ by hypothesis, so $A\mathbf{v}_j = A(B\mathbf{e}_j) = \mathbf{e}_j$. Hence, $B(A\mathbf{v}_j) = B\mathbf{e}_j = \mathbf{v}_j$, as desired. $\square$

B.2. Equivalence among various perspectives on linear subspaces. We initially defined a *linear subspace* of $\mathbf{R}^d$ to be the span of a finite collection of d-vectors. For example, the column space $C(A)$ of an $m \times n$ matrix A is the span of the n columns considered as vectors in $\mathbf{R}^m$, so it is a linear subspace of $\mathbf{R}^m$ due to its very definition.

In contrast, the null space $N(A)$ is defined in a very different manner: it is the collection of vectors $\mathbf{x} \in \mathbf{R}^n$ that satisfy $A\mathbf{x} = \mathbf{0}$, which amounts to m equations

$$\mathbf{a}_1 \cdot \mathbf{x} = 0, \;\; \mathbf{a}_2 \cdot \mathbf{x} = 0, \;\; \dots, \;\; \mathbf{a}_m \cdot \mathbf{x} = 0$$

in the n coordinates $x_1, \dots, x_n$ of $\mathbf{x}$, where $\mathbf{a}_i$ is the ith *row* of A. In other words, if we write the ith row of A as $\mathbf{a}_i = (a_{i1}, \dots, a_{in}) \in \mathbf{R}^m$ then $N(A)$ is the solution set to the system of m equations

$$\sum_{j=1}^n a_{1j}x_j = 0, \;\; \sum_{j=1}^n a_{2j}x_j = 0, \;\; \dots, \;\; \sum_{j=1}^n a_{mj}x_j = 0 \qquad \text{(B.2.1)}$$

in the n unknowns $x_1, \dots, x_n$.

It is not evident from this latter description that $N(A)$ is a linear subspace of $\mathbf{R}^n$. Let's explain what the issue is. On the one hand, the visualization we have in our head for each equation in the system is that its solution set is a "hyperplane" in $\mathbf{R}^n$ through the origin (assuming the equation isn't the silly case of having all coefficients equal to 0), generalizing the picture of a plane through the origin in the case $n = 3$. Thus, the geometric meaning of such a simultaneous system is akin to that of two different planes in $\mathbf{R}^3$ through the origin meeting along a line through the origin. In other words, our geometric experience in $\mathbf{R}^3$ strongly suggests that $N(A)$ "looks like" it ought to be a linear subspace (much as two different planes through the origin in $\mathbf{R}^3$ intersect in another linear subspace: a line). But can we *prove* it based on the actual *definition* of what a linear subspace is? In effect, we are trying to show that geometric and algebraic perspectives are equivalent to each other (which is the source of much of the power of linear algebra).

The task is to show that there is actually some collection of vectors $\mathbf{v}_1, \dots, \mathbf{v}_k \in N(A)$ so that the elements of $N(A)$ are precisely the linear combinations of the $\mathbf{v}_i$'s. In algebraic terms, this asks if there is some *finite* collection of solutions to the system (B.2.1) of m simultaneous equations so that the general solutions are obtained precisely by forming linear combinations of those finitely many solutions. When phrased that way in terms of the actual definition of the concept "linear subspace", its validity for $N(A)$ is not so apparent as a visual picture suggests. Fortunately, everything turns out nicely:

Theorem B.2.1. For any $m \times n$ matrix A, the null space $N(A)$ is a linear subspace of $\mathbf{R}^n$. Equivalently, for any m vectors $\mathbf{a}_1, \dots, \mathbf{a}_m \in \mathbf{R}^n$, the solution set of vectors $\mathbf{x} \in \mathbf{R}^n$ to the simultaneous equations

$$\mathbf{a}_1 \cdot \mathbf{x} = 0, \;\; \mathbf{a}_2 \cdot \mathbf{x} = 0, \;\; \dots, \;\; \mathbf{a}_m \cdot \mathbf{x} = 0$$

is a linear subspace of $\mathbf{R}^n$.

A consequence of this result is that for any linear subspace $V = \mathrm{span}(\mathbf{a}_1, \dots, \mathbf{a}_m) \subset \mathbf{R}^n$, its orthogonal complement is a linear subspace since $V^\perp$ is exactly the collection of $\mathbf{x}$'s as in Theorem B.2.1.

To understand what is going on in this theorem, we shall now step away from the specificity of null spaces (so as to cut down on the amount of notation) and formulate a few *properties* of any null space. It will turn out that *any* collection of vectors in $\mathbf{R}^m$ satisfying this short list of properties is a linear subspace! The starting point is the following temporary concept (whose terminology we have cooked up solely for the present discussion; it is *not* standard terminology outside this course) that captures the common features of linear subspaces and null spaces that were observed in Propositions 4.1.11 and 21.3.5.

Definition B.2.2. A collection W of vectors in $\mathbf{R}^n$ is called an *abstract subspace* if it satisfies the following conditions:

(i) $\mathbf{0} \in W$,
(ii) if $\mathbf{w}, \mathbf{w}' \in W$ then $\mathbf{w} + \mathbf{w}' \in W$,
(iii) if $\mathbf{w} \in W$ and c is any scalar then $c\mathbf{w} \in W$.

In other words: W contains the origin and is "preserved" under the operations of vector addition and scalar multiplication on itself.

Note that if we iterate conditions (ii) and (iii) repeatedly, together they are just saying that any linear combination of vectors in W again belongs to W. That is: for any $\mathbf{w}_1, \dots, \mathbf{w}_m \in W$, we have

$$\operatorname{span}(\mathbf{w}_1, \dots, \mathbf{w}_m) \subset W.$$

Example B.2.3. Every linear subspace V of $\mathbf{R}^n$ is an abstract subspace. This was already recorded in Proposition 4.1.11, but let's review the reasoning once again. By definition of "linear subspace", V is the span of some $\mathbf{v}_1, \dots, \mathbf{v}_k \in \mathbf{R}^n$. Thus,

$$\mathbf{0} = 0\,\mathbf{v}_1 + \dots 0\,\mathbf{v}_k \in V$$

(verifying Definition B.2.2(i) for V). Also, for any $\mathbf{w}, \mathbf{w}' \in W$ we have

$$\mathbf{w} = \sum_{j=1}^{k} a_j \mathbf{v}_j, \quad \mathbf{w}' = \sum_{j=1}^{k} b_j \mathbf{v}_j$$

for some scalars $a_1, \dots, a_k, b_1, \dots, b_k \in \mathbf{R}$ and hence

$$\mathbf{w} + \mathbf{w}' = \sum_{j=1}^{k} (a_j \mathbf{v}_j + b_j \mathbf{v}_j) = \sum_{j=1}^{k} (a_j + b_j)\mathbf{v}_j \in V$$

(verifying Definition B.2.2(ii) for V). Finally, for any $\mathbf{w} \in V$ and scalar c we have $\mathbf{w} = \sum_{j=1}^{k} a_j \mathbf{v}_j$ for some scalars $a_1, \dots, a_k \in \mathbf{R}$, so

$$c\,\mathbf{w} = \sum_{j=1}^{k} (ca_j)\mathbf{v}_j \in V$$

(verifying Definition B.2.2(iii) for V). ■

Here is a setting that may initially seem to be far-removed from linear algebra yet is a situation in which the concept of an "abstract subspace" very naturally emerges.

Example B.2.4. Consider the differential equation

$$y'' + 2y' - 3y = 0.$$

Such "second-order constant-coefficient" differential equations arise in the description of positions in many mechanical systems (where the first derivative corresponds to velocity and the second derivative corresponds to acceleration). An uninteresting solution to this is the constant function $y(x) = 0$. More interesting solutions are e^{-3x} and $(1/4)e^x$ (check these work).

Now something curious happens: the sum of these two explicit solutions is yet another solution. Indeed, for $f(x) = e^{-3x} + (1/4)e^x$ we have

$$\begin{aligned} f'' + 2f' - 3f &= (9e^{-3x} + (1/4)e^x) + 2(-3e^{-3x} + (1/4)e^x) - 3(e^{-3x} + (1/4)e^x) \\ &= (9e^{-3x} - 6e^{-3x} - 3e^{-3x}) + ((1/4)e^x + (2/4)e^x) - (3/4)e^x) \\ &= (9 - 6 - 3)e^{-3x} + ((1/4) + (2/4) - (3/4))e^x \\ &= 0. \end{aligned}$$

In fact, we can do much more than form the sum: *any* linear combination $f(x) = ae^{-3x} + b(1/4)e^x = ae^{-3x} + (b/4)e^x$ for $a, b \in \mathbf{R}$ is a solution!

Rather than grind out the algebra of plugging such an expression into the differential equation and observing magical cancellations, we make an even more striking observation: if $y_1(x), y_2(x)$ are *any* two solutions to the given differential equation then *any* linear combination $y(x) = c_1y_1(x) + c_2y_2(x)$ is also a solution. Indeed:

$$\begin{aligned} y'' + 2y' - 3y &= (c_1y_1 + c_2y_2)'' + 2(c_1y_1 + c_2y_2)' - 3(c_1y_1 + c_2y_2) \\ &= (c_1y_1'' + c_2y_2'') + 2(c_1y_1' + c_2y_2') - 3(c_1y_1 + c_2y_2) \\ &= (c_1y_1'' + c_2y_2'') + (2c_1y_1' + 2c_2y_2') - 3c_1y_1 - 3c_2y_2, \end{aligned}$$

and collecting together the terms involving a given scalar c_j turns this into

$$c_1(y_1'' + 2y_1' - 3y_1) + c_2(y_2'' + 2y_2' - 3y_2). \tag{B.2.2}$$

Aha, but $y_1(x)$ and $y_2(x)$ have been assumed to satisfy the original differential equation, which says exactly that

$$y_1'' + 2y_1' - 3y_1 = 0, \quad y_2'' + 2y_2' - 3y_2 = 0,$$

so (B.2.2) collapses to $c_1(0) + c_2(0) = 0$. Putting it all together, this says that the function $y(x) = c_1y_1(x) + c_2y_2(x)$ satisfies $y'' + 2y' - 3y = 0$, as desired.

What just happened? We have shown that the collection of *all* solutions to our initial differential equation satisfies conditions exactly like those in Definition B.2.2. In Math 53 you will explore the striking consequences of this link between differential equations and linear algebra. It leads to a compelling explanation for the fact the solutions $ae^{-3x} + (b/4)e^x$ with $a, b \in \mathbf{R}$ actually account for *all* solutions of the given differential equation, and it leads to a *systematic method* to use insights from linear algebra to determine all solutions to a much wider class of "constant-coefficient" differential equations. Even in the case of second-order equations (the type which arise frequently in physics and engineering applications), this is a rather illuminating perspective. ■

In most textbooks and courses on linear algebra, the somewhat abstract Definition B.2.2 is taken as the *definition* of "linear subspace" (of $\mathbf{R}^n$). We have taken the more geometric and concrete approach of defining "linear subspace" of $\mathbf{R}^n$ to be a span of finitely many vectors. We just saw that the more abstract concept includes the more geometric one as a special case. The crucial fact is that the seemingly more general concept of abstract subspace is in fact *exactly the same thing* as the concept of linear subspace: see Theorem B.2.6. The reason for introducing the seemingly more general notion is that it is sometimes easier to verify. Knowing these concepts agree opens the door to many nice consequences.

Another source of examples of abstract subspaces is null spaces:

Example B.2.5. If A is an $m \times n$ matrix then we claim that the null space $N(A)$ is an abstract subspace of $\mathbf{R}^n$. If $\mathbf{a}_1, \ldots, \mathbf{a}_m$ are the rows of A (considered as vectors in $\mathbf{R}^n$) then $N(A)$ consists of exactly those vectors $\mathbf{x} \in \mathbf{R}^n$ satisfying the m conditions

$$\mathbf{a}_1 \cdot \mathbf{x} = 0, \quad \mathbf{a}_2 \cdot \mathbf{x} = 0, \quad \ldots, \quad \mathbf{a}_m \cdot \mathbf{x} = 0.$$

We want to show that this collection of conditions on $\mathbf{x}$ satisfies the requirements in Definition B.2.2. This was already recorded as Proposition 21.3.5, but let's review the reasoning again. The vector $\mathbf{0} \in \mathbf{R}^n$ belongs to $N(A)$ since $\mathbf{0}$ is orthogonal to all $\mathbf{a}_j$'s (or because the matrix-vector product $A\mathbf{0}$ vanishes, which comes to the same thing). If $\mathbf{x}, \mathbf{x}' \in N(A)$ then $\mathbf{x} + \mathbf{x}' \in N(A)$ because for all $1 \leq j \leq n$ we have

$$\mathbf{a}_j \cdot (\mathbf{x} + \mathbf{x}') = \mathbf{a}_j \cdot \mathbf{x} + \mathbf{a}_j \cdot \mathbf{x}' = 0 + 0 = 0,$$

so $\mathbf{x} + \mathbf{x}' \in N(A)$. Finally, if $\mathbf{x} \in N(A)$ and $c \in \mathbf{R}$ then $c\mathbf{x} \in N(A)$ because for all $1 \leq j \leq m$ we have

$$\mathbf{a}_j \cdot (c\mathbf{x}) = c(\mathbf{a}_j \cdot \mathbf{x}) = c(0) = 0.$$

■

Here is the crux of the matter.

Theorem B.2.6. Every abstract subspace $V \subset \mathbf{R}^n$ is a linear subspace; i.e., V is the span of a finite collection of vectors in $\mathbf{R}^n$.

By Example B.2.5 we can apply this to $V = N(A)$, so Theorem B.2.1 is thereby established too.

PROOF. By definition, V contains $\mathbf{0}$. If it has no nonzero vectors then V is just the span of the single vector $\mathbf{0}$ and so is a linear subspace. Hence, we may and do assume V contains some nonzero vector $\mathbf{v}$. But V is an abstract subspace of $\mathbf{R}^n$, so V contains the line $\mathrm{span}(\mathbf{v})$. In the case $n = 1$ there is no room left for dimension reasons: we must have $V = \mathbf{R}^1$, again a linear subspace. This settles the case $n = 1$, so we now focus on the case $n > 1$.

Working up one step at a time for n, we may suppose the result is already known to hold for abstract subspaces of $\mathbf{R}^{n-1}$. In particular, we may suppose all abstract subspaces of $\mathbf{R}^{n-1}$ are known to be linear subspaces of $\mathbf{R}^{n-1}$. If V is an abstract subspace of $\mathbf{R}^n$ and every vector in V has vanishing nth coordinate then we can view V as living inside $\mathbf{R}^{n-1}$, so we would be done (i.e., V would be the span of finitely many vectors in $\mathbf{R}^{n-1}$ viewed as vectors in $\mathbf{R}^n$ by appending to 0 as a final coordinate of everything in sight). Thus, we can assume V contains some vector $\mathbf{v}$ whose final coordinate v_n is nonzero. Since V is an abstract subspace, it contains the vector $\mathbf{w} = (1/v_n)\mathbf{v}$ whose nth coordinate is equal to 1.

For any $\mathbf{x} \in V \subset \mathbf{R}^n$ with ith coordinate x_i, the vector $\mathbf{x} - x_n\mathbf{w}$ is a linear combination of $\mathbf{x}, \mathbf{w} \in V$, so $\mathbf{x} - x_n\mathbf{w} \in V$ by what it means to say that V is an abstract subspace. But since $\mathbf{w}$ has its final coordinate equal to 1, the final coordinate of $\mathbf{x} - x_n\mathbf{w}$ is equal to $x_n - x_n \cdot 1 = x_n - x_n = 0$. In other words, if we form the overlap

$$V' = \{\mathbf{v} = (x_1, \dots, x_n) \in V : x_n = 0\}$$

of V with the collection of vectors in $\mathbf{R}^n$ having vanishing final coordinate then the formula

$$\mathbf{x} = (\mathbf{x} - x_n\mathbf{w}) + x_n\mathbf{w}$$

expresses everything in V as a sum of something in V' and a scalar multiple of $\mathbf{w} \in V$. Moreover, by design $V' \subset V$ and $\mathbf{w} \in V$, so every vector of the form $\mathbf{v}' + c\mathbf{w}$ for $\mathbf{v}' \in V'$ and $c \in \mathbf{R}$ belongs to V by definition of V being an abstract subspace.

Thus, we can view V in the following way: it consists of *exactly* the vectors of the form $\mathbf{v}' + c\mathbf{w}$ for some $\mathbf{v}'$ belonging to the abstract subspace $V' \subset \mathbf{R}^n$ and some scalar c. But everything in V' has vanishing final coordinate, so by ignoring that we can view V' inside $\mathbf{R}^{n-1}$ where it is certainly an abstract subspace (as V' is such inside $\mathbf{R}^n$). The gain is that we already know that abstract subspaces of $\mathbf{R}^{n-1}$ are linear subspaces, so V' is the span of some collection of vectors $\mathbf{v}'_1, \dots, \mathbf{v}'_k \in \mathbf{R}^{n-1}$. Viewing $\mathbf{v}'_1, \dots, \mathbf{v}'_k$ as n-vectors by appending an nth coordinate equal to 0, we conclude that the vectors in V are precisely the n-vectors of the form

$$a_1\mathbf{v}'_1 + \cdots + a_k\mathbf{v}'_k + c\mathbf{w}$$

for scalars $a_1, \ldots, a_k, c$. This says

$$V = \text{span}(\mathbf{v}_1', \ldots, \mathbf{v}_k', \mathbf{w}),$$

so V is a span of a finite set of vectors in $\mathbf{R}^n$, which is what we wanted to show! ☐

Since the null space $N(A)$ of an $m \times n$ matrix A is now known to be a linear subspace, it has a dimension (smallest size of a spanning set). This allows us to state a result that was seen in Section 21.6 to lie at the heart of the equality of row rank and column rank (Theorem 21.6.1).

Theorem B.2.7 (Rank–Nullity Theorem). For any $m \times n$ matrix A, $\dim N(A) + \dim C(A) = n$.

Applying this to the $n \times m$ matrix $A^\top$ gives $\dim N(A^\top) + \dim C(A^\top) = m$, which is the version that arose in our discussion at the end of Section 21.6.

PROOF. If $C(A) = \{\mathbf{0}\}$ then A is the zero matrix, so $N(A) = \mathbf{R}^n$ and there is nothing to do. Hence, we may suppose $C(A)$ is nonzero. Let $d = \dim C(A) > 0$, and pick a basis $\mathbf{b}_1, \ldots, \mathbf{b}_d$ of $C(A)$. By the very meaning of the column space, for each $\mathbf{b}_j$ we can pick a solution $\mathbf{v}_j \in \mathbf{R}^n$ to $A\mathbf{x} = \mathbf{b}_j$; i.e., $A\mathbf{v}_j = \mathbf{b}_j$.

The first observation to make is that the n-vectors $\mathbf{v}_1, \ldots, \mathbf{v}_d$ are linearly independent. That is, if $c_1, \ldots, c_d$ are scalars for which $c_1\mathbf{v}_1 + \cdots + c_d\mathbf{v}_d = \mathbf{0}$ then we claim that the c_j's all vanish. By applying the matrix-vector product against A to both sides, we get

$$A(c_1\mathbf{v}_1 + \cdots + c_d\mathbf{v}_d) = A\mathbf{0} = \mathbf{0}$$

in $\mathbf{R}^m$. The left side equal to $c_1A\mathbf{v}_1 + \cdots + c_dA\mathbf{v}_d = c_1\mathbf{b}_1 + \cdots + c_d\mathbf{b}_d$, so $\sum_{j=1}^d c_j\mathbf{b}_j = \mathbf{0}$. But the $\mathbf{b}_j$'s are linearly independent (as they constitute a basis of $C(A)$ by design), so this vanishing linear combination forces all coefficients c_j to vanish, as desired.

Now that the $\mathbf{v}_j$'s are known to be linearly independent, let's see how any $\mathbf{v} \in \mathbf{R}^n$ is related to the $\mathbf{v}_j$'s. The m-vector $A\mathbf{v}$ certainly belongs to $C(A)$, which in turn is spanned by the $\mathbf{b}_j$'s (since the $\mathbf{b}_j$'s are a basis of $C(A)$ by design), so there are scalars $c_1, \ldots, c_d$ for which

$$A\mathbf{v} = c_1\mathbf{b}_1 + \cdots + c_d\mathbf{b}_d = c_1A\mathbf{v}_1 + \cdots + c_dA\mathbf{v}_d = A(c_1\mathbf{v}_1 + \cdots + c_d\mathbf{v}_d).$$

Comparing the left and right sides, we have two vectors giving the same output under A. But whenever $A\mathbf{v} = A\mathbf{v}'$ for n-vectors $\mathbf{v}$ and $\mathbf{v}'$, we have $A(\mathbf{v} - \mathbf{v}') = A\mathbf{v} - A\mathbf{v}' = \mathbf{0}$, so $\mathbf{v} - \mathbf{v}' \in N(A)$. Hence, if we define

$$\mathbf{w} = \mathbf{v} - (c_1\mathbf{v}_1 + \cdots + c_d\mathbf{v}_d)$$

then $\mathbf{w} \in N(A)$ and

$$\mathbf{v} = \mathbf{w} + c_1\mathbf{v}_1 + \cdots + c_d\mathbf{v}_d. \tag{B.2.3}$$

In words, everything in $\mathbf{R}^n$ is obtained by combining the null space of A and the linearly independent $\mathbf{v}_j$'s.

We argue separately depending on whether or not $N(A) = \{\mathbf{0}\}$. First assume $N(A) = \{\mathbf{0}\}$, so $\dim N(A) = 0$ and our desired dimension formula is $\dim C(A) \overset{?}{=} n$, which is to say $d \overset{?}{=} n$. In other words, in this case we have to show $d = n$. Well, $\mathbf{w}$ above vanishes since it lies inside the null space $N(A)$ that we are now assuming is just the zero vector, and so (B.2.3) says that the arbitrary $\mathbf{v}$ is in the span of the $\mathbf{v}_j$'s. Hence, the linearly independent $\mathbf{v}_j$'s span the entirety of $\mathbf{R}^n$. But a linearly independent spanning set is the same thing as a basis, so the number d of such $\mathbf{v}_j$'s must equal $\dim \mathbf{R}^n = n$, as desired.

Now suppose $N(A)$ is not just the zero vector. Being a nonzero linear subspace of $\mathbf{R}^n$, we can pick a basis $\mathbf{w}_1, \ldots, \mathbf{w}_r$ of $N(A)$, where $r = \dim N(A)$. We are going to show that the collection of $d + r$ vectors

$$\mathbf{v}_1, \ldots, \mathbf{v}_d, \mathbf{w}_1, \ldots, \mathbf{w}_r$$

is a basis for $\mathbf{R}^n$, so $n = d + r$, which is the desired result since $d = \dim C(A)$ by definition. This is a spanning set for $\mathbf{R}^n$ because (B.2.3) shows that every $\mathbf{v} \in \mathbf{R}^n$ can be written as

$$\mathbf{v} = \mathbf{w} + c_1\mathbf{v}_1 + \cdots + c_d\mathbf{v}_d = a_1\mathbf{w}_1 + \cdots + a_r\mathbf{w}_r + c_1\mathbf{v}_1 + \cdots + c_d\mathbf{v}_d$$

(where $\sum_{j=1}^{r} a_j\mathbf{w}_j$ is an expression for $\mathbf{w} \in N(A)$ as a linear combination of the $\mathbf{w}_j$'s that span $N(A)$ by design). Hence, if this collection of $d + r$ vectors is also linearly independent then it would be a basis of its span $\mathbf{R}^n$, which is what we are aiming to show.

Finally, it remains to check that the collection of $\mathbf{v}_i$'s and $\mathbf{w}_j$'s is linearly independent. To that end, we suppose for some scalars $c_1, \ldots, c_d, a_1, \ldots, a_r$ that

$$c_1\mathbf{v}_1 + \cdots + c_d\mathbf{v}_d + a_1\mathbf{w}_1 + \cdots + a_r\mathbf{w}_r = \mathbf{0} \tag{B.2.4}$$

and we just need to deduce that all of these scalar coefficients vanish. First we will show that the c_i's all vanish, and then that the a_j's all vanish. Applying the matrix-vector product against A to both sides gives

$$A(c_1\mathbf{v}_1 + \cdots + c_d\mathbf{v}_d + a_1\mathbf{w}_1 + \cdots + a_r\mathbf{w}_r) = A\mathbf{0} = \mathbf{0},$$

and the left side (by "linearity") is equal to

$$c_1A\mathbf{v}_1 + \cdots + c_dA\mathbf{v}_d + a_1A\mathbf{w}_1 + \cdots + a_rA\mathbf{w}_r = c_1\mathbf{b}_1 + \cdots + c_d\mathbf{b}_d + a_1\mathbf{0} + \cdots + a_r\mathbf{0} = c_1\mathbf{b}_1 + \cdots + c_d\mathbf{b}_d.$$

(Here we have used that each $\mathbf{w}_j$ belongs to $N(A)$ by design, and that $A\mathbf{v}_i = \mathbf{b}_i$ by design of $\mathbf{v}_i$.) Hence, we have

$$c_1\mathbf{b}_1 + \cdots + c_d\mathbf{b}_d = \mathbf{0}.$$

But the $\mathbf{b}_i$'s are linearly independent (they constitute a basis of $C(A)$ by design), so this forces the coefficients $c_1, \ldots, c_d$ to vanish. Plugging this conclusion into (B.2.4) yields

$$a_1\mathbf{w}_1 + \cdots + a_r\mathbf{w}_r = \mathbf{0}.$$

By design the $\mathbf{w}_j$'s are linearly independent since they constitute a basis of $N(A)$, so this forces the a_j's to vanish too. ☐

Corollary B.2.8. Let be A an $n \times n$ matrix.

(a) If A is not invertible then $N(A)$ contains a nonzero vector. In particular, whenever the vector equation $A\mathbf{x} = \mathbf{b}$ has a solution $\mathbf{x}_0$, it has more than one solution (namely $\mathbf{x}_0 + \mathbf{v}$ for any nonzero $\mathbf{v} \in N(A)$), so it *never* happens for such A and some $\mathbf{b}$ that $A\mathbf{x} = \mathbf{b}$ has exactly one solution.

(b) If an $n \times n$ matrix A has vanishing null space then it is invertible.

PROOF. Once the first assertion in (a) is proved, the rest of (a) is immediate. Likewise, (b) would also follow since by (a) the vanishing of $N(A)$ never happens in the non-invertible case (so when $N(A) = \{\mathbf{0}\}$ it follows that A cannot be non-invertible, so it must be invertible). We therefore focus on the first assertion in (a).

By Theorem 18.3.3, the invertibility of an $n \times n$ matrix M amounts to the equation $M\mathbf{x} = \mathbf{b}$ having a unique solution for *every* n-vector $\mathbf{b}$. Since A is *not* invertible, this tells us that for *some* n-vector $\mathbf{b}_0$ the equation $A\mathbf{x} = \mathbf{b}_0$ must fail to "have a unique solution". That means one of two things must happen: either (i) there is no solution *or* (ii) there is a solution but actually more than one. If case (ii) is what occurs for $\mathbf{b}_0$ then we are done because Proposition 21.3.10 then tells us that for any two different solutions to $A\mathbf{x} = \mathbf{b}_0$, their *nonzero* difference lies in the null space (so $N(A)$ is nonzero, as desired). But how do we rule out the possibility that for every $\mathbf{b}_0$ for which $A\mathbf{x} = \mathbf{b}_0$ fails to have a unique solution (i.e., either no solution or more than one solution), it is always case (i) that occurs?

Let's suppose it is always case (i) that occurs, so the vector equation $A\mathbf{x} = \mathbf{b}$ for every n-vector $\mathbf{b}$ has at most one solution (i.e., either no solution or just one solution). From this supposition we will deduce an inconsistency, so in fact such a situation never could have arisen. Since case (ii) is never occurring, we must have $N(A) = \mathbf{0}$ since otherwise $A\mathbf{x} = \mathbf{0}$ falls into case (ii). We conclude by Theorem B.2.7 that $\dim C(A) = n - \dim N(A) = n$, so the linear subspace $C(A) \subset \mathbf{R}^n$ has dimension n and hence $C(A) = \mathbf{R}^n$ (Theorem 4.2.8). Thus, *every* $\mathbf{b} \in \mathbf{R}^n$ lies in the column space, which is to say $A\mathbf{x} = \mathbf{b}$ has a solution for *every* $\mathbf{b}$.

But we have assumed that the vector equation $A\mathbf{x} = \mathbf{b}$ (for any $\mathbf{b}$ at all) has *at most* one solution! Since we have deduced (in our situation) that there is always some solution, we conclude that there is always exactly one solution. The property for an $n \times n$ matrix A that the equation $A\mathbf{x} = \mathbf{b}$ has exactly one solution for every n-vector $\mathbf{b}$ is one of the equivalent definitions of the invertibility of the matrix (see Proposition 18.1.5(a) and Definition 18.1.6). We have assumed A is not invertible, so we have reached an inconsistency. Hence, our initial assumption that case (ii) never occurs for a non-invertible A is false, and so (as explained above) we are done. ☐

B.3. Geometric proof of the Spectral Theorem. Our aim here is to prove Theorem 24.1.4 using geometric ideas. We require a refinement of Theorem 20.1.4: that result (whose proof was a direct calculation in matrix algebra) says that if A is an $n \times n$ matrix then for any $\mathbf{v}, \mathbf{w} \in \mathbf{R}^n$ we have $(A\mathbf{v}) \cdot \mathbf{w} = \mathbf{v} \cdot (A^\top \mathbf{w})$, so in particular if A is symmetric (i.e., $A^\top = A$) then $(A\mathbf{v}) \cdot \mathbf{w} = \mathbf{v} \cdot (A\mathbf{w})$ for all $\mathbf{v}, \mathbf{w} \in \mathbf{R}^n$. The connection to symmetry goes the other way too:

Proposition B.3.1. An $n \times n$ matrix A is symmetric precisely when $(A\mathbf{v}) \cdot \mathbf{w} = \mathbf{v} \cdot (A\mathbf{w})$ for all $\mathbf{v}, \mathbf{w} \in \mathbf{R}^n$.

PROOF. If A is symmetric then the desired identity holds; this is part of Theorem 20.1.4. For the opposite implication, suppose $(A\mathbf{v}) \cdot \mathbf{w} = \mathbf{v} \cdot (A\mathbf{w})$ for all $\mathbf{v}, \mathbf{w} \in \mathbf{R}^n$; we want to deduce that A is necessarily symmetric. Letting $\mathbf{v} = \mathbf{e}_i$ and $\mathbf{w} = \mathbf{e}_j$, we have $(A\mathbf{e}_i) \cdot \mathbf{e}_j = \mathbf{e}_i \cdot (A\mathbf{e}_j)$. The left side is the jth entry of the ith column $A\mathbf{e}_i$ of A, which is to say it equals a_{ji}. The right side is the ith entry of the jth column, which is to say it equals a_{ij}. We conclude that $a_{ij} = a_{ji}$ for all i, j, so A is symmetric. ☐

Remark B.3.2. As a nice application of Proposition B.3.1, we can explain the symmetry of the $n \times n$ matrix A for $\mathbf{Proj}_V : \mathbf{R}^n \to \mathbf{R}^n$ for any linear subspace V of $\mathbf{R}^n$ (which was mentioned without proof in Proposition 20.3.10). By Proposition B.3.1, it suffices to show that for any $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$ we have $(A\mathbf{x}) \cdot \mathbf{y} = \mathbf{x} \cdot (A\mathbf{y})$. By Theorem 6.2.1, the differences $\mathbf{x} - \mathbf{Proj}_V(\mathbf{x})$ and $\mathbf{y} - \mathbf{Proj}_V(\mathbf{y})$ are each orthogonal to *everything* in V, so $\mathbf{x} - A\mathbf{x}$ and $\mathbf{y} - A\mathbf{y}$ are each orthogonal to everything in V. But the projections $A\mathbf{y}$ and $A\mathbf{x}$ belong to V, so

$$(\mathbf{x} - A\mathbf{x}) \cdot (A\mathbf{y}) = 0, \quad (\mathbf{y} - A\mathbf{y}) \cdot (A\mathbf{x}) = 0.$$

These respectively say $\mathbf{x} \cdot (A\mathbf{y}) - (A\mathbf{x}) \cdot (A\mathbf{y}) = 0$ and $\mathbf{y} \cdot (A\mathbf{x}) - (A\mathbf{y}) \cdot (A\mathbf{x}) = 0$, so

$$\mathbf{x} \cdot (A\mathbf{y}) = (A\mathbf{x}) \cdot (A\mathbf{y}) = (A\mathbf{y}) \cdot (A\mathbf{x}) = \mathbf{y} \cdot (A\mathbf{x}) = (A\mathbf{x}) \cdot \mathbf{y}.$$

The equality of the outer terms is exactly what we needed to show.

Here is a proof of the Spectral Theorem in 3 steps.

Step 1. The first observation is that it suffices to show in general that a symmetric $n \times n$ matrix A has *some* eigenvector, for all n. Indeed, granting this fact in general (for all n), let's see how to deduce the full result, first building the $\mathbf{w}_j$'s. The case $n = 1$ is obvious, so we may now assume $n > 1$. Let $\mathbf{v} \in \mathbf{R}^n$ be an eigenvector of A (the existence of such $\mathbf{v}$ in general is what we are temporarily taking as known). Let λ be the corresponding eigenvalue. Let $H = \mathbf{v}^\perp$ be the hyperplane perpendicular to the line spanned by $\mathbf{v}$, so (applying Theorem 19.2.5 to the line $\mathrm{span}(\mathbf{v})$) we have $\dim H = n - 1 < n$.

We claim that the linear transformation $T_A : \mathbf{R}^n \to \mathbf{R}^n$ corresponding to A (i.e., $T_A(\mathbf{x}) = A\mathbf{x}$ for all $\mathbf{x} \in \mathbf{R}^n$) carries H into itself. That is, we claim that if $\mathbf{w} \in H$ then $A\mathbf{w} \in H$. By definition of H as $\mathbf{v}^\perp$, we use the symmetry of A and Proposition B.3.1 (or really Theorem 20.1.4) to compute

$$(A\mathbf{w}) \cdot \mathbf{v} = \mathbf{w} \cdot (A\mathbf{v}) = \mathbf{w} \cdot (\lambda\mathbf{v}) = \lambda(\mathbf{w} \cdot \mathbf{v}) = \lambda(0) = 0,$$

so $A\mathbf{w} \in \mathbf{v}^{\perp} = H$.

Now we use Gram–Schmidt to build an *orthonormal* basis $\{\mathbf{h}_1, \dots, \mathbf{h}_{n-1}\}$ of H. Recall that this amounts to building an orthogonal basis and then dividing each basis vector by its length. Since the basis vectors are orthogonal with unit length, this basis makes H "look like" $\mathbf{R}^{n-1}$: vectors in H are uniquely written as $\mathbf{h} = \sum_{i=1}^{n-1} x_i \mathbf{h}_i$ for scalars $x_1, \dots, x_{n-1}$ and the dot product on H (inherited from $\mathbf{R}^n$) looks like with the usual dot product on $\mathbf{R}^{n-1}$:

$$\begin{aligned}(\sum_{i=1}^{n-1} x_i \mathbf{h}_i) \cdot (\sum_{j=1}^{n-1} y_j \mathbf{h}_j) &= \sum_{1 \le i,j \le n-1} (x_i \mathbf{h}_i) \cdot (y_j \mathbf{h}_j) \\ &= \sum_{1 \le i,j \le n-1} (x_i y_j) \mathbf{h}_i \cdot \mathbf{h}_j \\ &= \sum_{i=1}^{n} x_i y_i\end{aligned}$$

(where the final equality uses that $\mathbf{h}_i \cdot \mathbf{h}_j$ vanishes for $i \neq j$ and equals 1 for $i = j$, due to $\{\mathbf{h}_1, \dots, \mathbf{h}_{n-1}\}$ being chosen as orthonormal; i.e., pairwise orthogonal unit vectors). The point is that the final expression in this chain of equalities is the usual formula for the dot product on $\mathbf{R}^{n-1}$.

The preceding argument shows that the linear transformation $T_A : \mathbf{R}^n \to \mathbf{R}^n$ restricts to a linear transformation $T_A : H \to H$, and this in turn corresponds to some $(n-1) \times (n-1)$ matrix B under the identification of H with $\mathbf{R}^{n-1}$ that we made using the $\mathbf{h}_i$'s (i.e., for each $\mathbf{h} \in H$ we have $\mathbf{h} = \sum_{i=1}^{n-1} x_i \mathbf{h}_i$ and computations of linear combinations and dot products with such $\mathbf{h}$'s is given by the usual formulas in terms of the associated coefficients $(x_1, \dots, x_{n-1}) \in \mathbf{R}^{n-1}$). We claim that B is *symmetric*. For this we use the criterion from Proposition B.3.1: since the identification of $\mathbf{R}^{n-1}$ with H respects notions of dot product on each, the symmetry criterion $(B\mathbf{x}) \cdot \mathbf{x}' = \mathbf{x} \cdot (B\mathbf{x}')$ for any $\mathbf{x}, \mathbf{x}' \in \mathbf{R}^{n-1}$ amounts to the assertion that $T_A(\mathbf{v}) \cdot \mathbf{v}' = \mathbf{v} \cdot T_A(\mathbf{v}')$ for all $\mathbf{v}, \mathbf{v}' \in H$. But this is a special case of the general identity $(A\mathbf{v}) \cdot \mathbf{v}' = \mathbf{v} \cdot (A\mathbf{v}')$ for all $\mathbf{v}, \mathbf{v}' \in \mathbf{R}^n$ (let alone $\mathbf{v}, \mathbf{v}' \in H$), which holds because A is symmetric.

If the Spectral Theorem were known for $(n-1) \times (n-1)$ symmetric matrices then we could apply it to the symmetric $(n-1) \times (n-1)$ matrix B to get an orthonormal basis of eigenvectors for B in $\mathbf{R}^{n-1}$. But eigenvectors for B in $\mathbf{R}^{n-1}$ correspond exactly to eigenvectors for A in H. Indeed, if

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} \in \mathbf{R}^{n-1}$$

and $\mathbf{h} = \sum_{i=1}^{n-1} x_i \mathbf{h}_i$ is the corresponding vector in H then by definition of B we know that $B\mathbf{x} \in \mathbf{R}^{n-1}$ corresponds to $A\mathbf{h} \in H$, so the eigenvector condition $B\mathbf{x} = \mu\mathbf{x}$ for a scalar μ is equivalent to the condition $A\mathbf{h} = \mu\mathbf{h}$. Since the usual dot product on $\mathbf{R}^{n-1}$ matches the dot product on H, an orthonormal basis of eigenvectors for B in $\mathbf{R}^{n-1}$ would thereby yield an orthonormal basis of H consisting of eigenvectors for A. When such a latter basis is combined with the unit eigenvector $\mathbf{v}$ for A that is orthogonal to the hyperplane H (due to how H was defined), we would get an orthonormal collection of n eigenvectors for A in $\mathbf{R}^n$. Any collection of n mutually orthogonal unit vectors in $\mathbf{R}^n$ has span that is n-dimensional (Theorem 5.2.2) and hence coincides with $\mathbf{R}^n$ (Theorem 4.2.8).

Thus, *if* it is known that symmetric square matrices of every size always admit *some* eigenvector then via induction on n we obtain the existence of an orthogonal basis of eigenvectors for symmetric matrices of every size. Moreover, for $\mathbf{w}_1, \dots, \mathbf{w}_n$ as in the statement of the Spectral Theorem and λ_j the eigenvalue for $\mathbf{w}_j$, let's check that the λ_j's are the only eigenvalues for A. Suppose $A\mathbf{w} = \lambda\mathbf{w}$ for some $\mathbf{w} \neq \mathbf{0}$ and

$\lambda \in \mathbf{R}$. By writing $\mathbf{w} = c_1\mathbf{w}_1 + \cdots + c_n\mathbf{w}_n$ for some scalars $c_1, \ldots, c_n$, we have

$$A\mathbf{w} = c_1(A\mathbf{w}_1) + \cdots + c_n(A\mathbf{w}_n) = c_1(\lambda_1\mathbf{w}_1) + \cdots + c_n(\lambda_n\mathbf{w}_n) = (\lambda_1c_1)\mathbf{w}_1 + \cdots + (\lambda_nc_n)\mathbf{w}_n$$

and $\lambda\mathbf{w} = (\lambda c_1)\mathbf{w}_1 + \cdots + (\lambda c_n)\mathbf{w}_n$. Since $\mathbf{w}_1, \ldots, \mathbf{w}_n$ is a basis, the coefficients λ_jc_j and λc_j of $\mathbf{w}_j$ in these two expressions for the vector $A\mathbf{w} = \lambda\mathbf{w}$ must be the same: $\lambda_jc_j = \lambda c_j$ for every j. But *some* c_{j_0} is nonzero since $\mathbf{w} \neq \mathbf{0}$, so we can cancel c_{j_0} to conclude that $\lambda_{j_0} = \lambda$; this says that λ is equal to one of the λ_j's, as desired.

Remark B.3.3. The geometric method used above to bootstrap from the existence of one eigenvector to the existence of more eigenvectors via considering the hyperplane orthogonal to an "eigenline" is not merely an idea with theoretical significance. The *exact same* technique underlies what is actually done in practice to transform an optimization algorithm for computing the biggest eigenvalue of a symmetric matrix (see Remark B.3.4) into an algorithm for directly computing the k biggest singular values in the SVD of an arbitrary matrix, for whatever k we wish (e.g., $k = 10$ or $k = 50$).

The tremendous practical importance of this is addressed in Remark 27.3.10, and it illustrates that in mathematics, theoretical and practical insights go hand in hand: when one understands the concepts behind why a result holds, it is often possible to build upon such insights to develop efficient practical algorithms.

Step 2. We are left with the task of proving for each n that every symmetric $n \times n$ matrix A has *some* eigenvector. In order to motivate how we will do this, let's admit the Spectral Theorem for a moment, and consider how to "extract" an eigenvalue from A. We can't hope to give an explicit formula for an eigenvalue, but we'll come quite close.

Granting temporarily (for motivational purposes) that there is an orthonormal basis $\mathbf{v}_1, \ldots, \mathbf{v}_n$ of (unit) eigenvectors of A for $\mathbf{R}^n$, by (24.2.2) the quadratic form $q_A(\mathbf{v}) = \mathbf{v} \cdot (A\mathbf{v}) = \mathbf{v}^\top A\mathbf{v}$ is expressed in $\mathbf{v}_j$-coordinates as

$$q_A(\sum_{i=1}^n x_i\mathbf{v}_i) = \sum_{i=1}^n \lambda_i x_i^2.$$

Rearrange the $\mathbf{v}_i$'s if necessary so that

$$\lambda_1 \leq \cdots \leq \lambda_n.$$

We define $\lambda_{\max} = \lambda_n$, so this is the largest eigenvalue of A (which might be negative). Here is the key observation: if $\mathbf{v} = \sum x_i\mathbf{v}_i$ then since $\|\mathbf{v}\|^2 = \sum_{i=1}^n x_i^2$ (as $\mathbf{v}_1, \ldots, \mathbf{v}_n$ is an orthonormal basis) we have

$$q_A(\mathbf{v}) = \sum_{i=1}^n \lambda_i x_i^2 \leq \sum_i \lambda_{\max} x_i^2 = \lambda_{\max}\|\mathbf{v}\|^2,$$

with equality achieved precisely when $x_i = 0$ whenever $\lambda_i < \lambda_{\max}$. That is, equality is achieved precisely when $\mathbf{v}$ is in the span of the $\mathbf{v}_i$'s with $\lambda_i = \lambda_{\max}$, which implies $\mathbf{v}$ is an eigenvector for A with eigenvalue $\lambda_{\max}$ (a nonzero linear combination of eigenvectors with the *same* eigenvalue is again an eigenvector with that same eigenvalue; beware that in contrast, a sum of eigenvectors having pairwise distinct eigenvalues is never an eigenvector).

In other words, for $\mathbf{v} \in \mathbf{R}^n$ with $\mathbf{v} \neq \mathbf{0}$ we have shown

$$\frac{q_A(\mathbf{v})}{\|\mathbf{v}\|^2} \leq \lambda_{\max} \tag{B.3.1}$$

with equality actually attained for some *eigenvectors* with eigenvalue $\lambda_{\max}$; i.e., for some *nonzero* solutions to $A\mathbf{v} = \lambda_{\max}\mathbf{v}$. (The left side of (B.3.1) is sometimes called a *Rayleigh quotient* for A.) Any eigenvector can be scaled to become a unit vector without affecting its property of being an eigenvector, nor affecting the corresponding eigenvalue. Thus, $q_A(\mathbf{v}) \leq \lambda_{\max}$ for unit vectors $\mathbf{v}$, with equality holding for some unit eigenvectors with eigenvalue $\lambda_{\max}$. (The Min-max Theorem gives a formula for the kth largest eigenvalue: if $m_k(W)$ denotes the *minimal* value of $q_A(\mathbf{v})$ for unit vectors $\mathbf{v}$ in a k-dimensional subspace W in $\mathbf{R}^n$ then λ_k is the *maximal* value of $m_k(W)$ as W varies. For $k = 1$ this recovers the above recipe for $\lambda_{\max}$.)

Remark B.3.4. The preceding geometric interpretation of the maximal eigenvalue as a maximal value for q_A on unit vectors is not just an idea of theoretical value: it underlies the numerical algorithm by which one actually *computes* the biggest eigenvalue and a corresponding eigenvector to very high accuracy on a computer! (One such algorithm is called "Rayleigh quotient iteration".)

Step 3. Now *forget* any assumption that A actually has an eigenvector. The preceding analysis, especially in Step 2, *motivates* us to define the function $q_A : S = \{\mathbf{x} \in \mathbf{R}^n : \|\mathbf{x}\| = 1\} \to \mathbf{R}$ on the "unit sphere" S in $\mathbf{R}^n$ by the formula $q_A(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x}$. We want to show that q_A actually *attains* a maximum value at some point $\mathbf{v}_0$ on the unit sphere S, and then directly prove that this $\mathbf{v}_0$ actually is an eigenvector for A (with eigenvalue equal to the maximum value of q_A on S).

This idea immediately runs into several issues:

- How do we know that q_A is even bounded on S? (Our earlier study of its behavior was predicated on the existence of an orthonormal basis of eigenvectors, which is what we are trying to prove.)
- Even once the boundedness is known, how can we prove a maximum value is actually *attained* by q_A at some point $\mathbf{v}_0$ on the unit sphere S? (Our earlier analysis giving the existence of a point where a maximum is attained was conditional on assuming the existence of eigenvectors for A, a result we are presently trying to prove.)
- Finally, even if we can show that a maximum value λ_A is attained by q_A at some point $\mathbf{v}_0$ on the unit sphere S, how are we going to prove $A\mathbf{v}_0 = \lambda_A\mathbf{v}_0$?

We now take care of all of these issues. The idea for the first two issues is that one should apply a variant of the Extreme Value Theorem from single-variable calculus. That result says that if $f : [a, b] \to \mathbf{R}$ is a continuous function to $\mathbf{R}$ on a closed bounded interval then f attains maximal and minimal values. We haven't defined what "continuity" means for a function $f : \mathbf{R}^n \to \mathbf{R}$, but it is rather plausible that under any reasonable definition of such continuity, a function such as q_A that is a (quadratic) polynomial expression in $x_1, \ldots, x_n$ has to be continuous. Furthermore, the Extreme Value Theorem does indeed carry over to continuous functions on regions like spheres $\{\|\mathbf{x}\| = r\}$ and balls $\{\|\mathbf{x}\| \le r\}$ in $\mathbf{R}^n$, as you'll learn if you take a course such as Math 115 or 171 (the key concept is called "compactness"). The upshot is that an appropriate generalization of the Extreme Value Theorem ensures that q_A attains a maximal value on S, which we will denote as λ_A.

The most serious issue is this: if $\mathbf{v}_0 \in S$ is a unit vector at which q_A attains its maximal value λ_A on S, why does $A\mathbf{v}_0 = \lambda_A\mathbf{v}_0$? Note that if $A\mathbf{v}_0 = \lambda\mathbf{v}_0$ for *some* scalar λ then $\lambda_A = q_A(\mathbf{v}_0) = \mathbf{v}_0 \cdot A\mathbf{v}_0 = \mathbf{v}_0 \cdot (\lambda\mathbf{v}_0) = \lambda(\mathbf{v}_0 \cdot \mathbf{v}_0) = \lambda\|\mathbf{v}_0\|^2 = \lambda$, so necessarily $\lambda = \lambda_A$ and we'd be done. So our problem is the same as showing that any $\mathbf{v}_0 \in S$ at which q_A attains a maximum must be an eigenvector of A (without needing to know in advance that the eigenvalue is λ_A; this follows automatically, as we just showed).

For this we will give two methods: one based on Lagrange multipliers from Chapter 12 (the eigenvalue will turn out to be the multiplier), and another that is more self-contained but modeled on ideas in the *proof* (in Section 12.5) of the main theorem on Lagrange multipliers (Theorem 12.2.1).

Method 1. The unit vector $\mathbf{v}_0$ is a solution to a constrained optimization problem: it optimizes the function $f(\mathbf{x}) = q_A(\mathbf{x}) = \mathbf{x} \cdot (A\mathbf{x})$ subject to the constraint that the function $g(\mathbf{x}) = \mathbf{x} \cdot \mathbf{x} = \sum_{j=1}^n x_j^2$ is equal to 1 (this says $\mathbf{x} \in S$). So the main theorem on Lagrange multipliers (Theorem 12.2.1) tells us that at $\mathbf{v}_0$ either the gradient ∇g vanishes or ∇f is a scalar multiple of ∇g.

But the gradient $(\nabla g)(\mathbf{x})$ is the vector whose jth entry is $\partial g/\partial x_j = 2x_j$, so $(\nabla g)(\mathbf{x}) = 2\mathbf{x}$, and this only vanishes at the origin, which is not on the unit sphere S. Hence, $(\nabla f)(\mathbf{v}_0) = \lambda(\nabla g)(\mathbf{v}_0) = 2\lambda\mathbf{v}_0$ for some scalar λ. We will now show that $(\nabla f)(\mathbf{x}) = 2A\mathbf{x}$ for all $\mathbf{x} \in \mathbf{R}^n$, so for $\mathbf{x} = \mathbf{v}_0$ we'd get

$2A\mathbf{v}_0 = 2\lambda\mathbf{v}_0$, exactly the eigenvector condition "$A\mathbf{v}_0 = \lambda\mathbf{v}_0$" that we wanted to establish! (When $n = 1$, the asserted identity $(\nabla f)(\mathbf{x}) = 2A\mathbf{x}$ is the fact that the function $f(x) = ax^2$ has derivative $2ax$.)

It remains to compute the gradient of $f(\mathbf{x}) = \mathbf{x} \cdot (A\mathbf{x})$. As a quadratic expression in the x_i's, the variable x_j appears in $f(\mathbf{x})$ in two ways: there is the contribution $a_{jj}x_j^2$ from the diagonal of A and the contributions $2a_{ij}x_ix_j$ from the common off-diagonal entry a_{ij} and $a_{ji} = a_{ij}$ in the ij and ji positions in A for each $i \neq j$. Applying the partial derivative with respect to x_j, the diagonal contribution gives $2a_{jj}x_j$ and the off-diagonal contributions give $2a_{ij}x_i$ for each $i \neq j$. Adding these all together, we get $\sum_{i=1}^{n} 2a_{ij}x_i$ as the jth entry of the vector $(\nabla f)(\mathbf{x})$. But this sum is twice the jth entry of the vector $A\mathbf{x}$, so $(\nabla f)(\mathbf{x})$ and $2A\mathbf{x}$ have the same jth entry for all j and hence these vectors are equal, as desired.

Method 2. Since $\lambda_A = q_A(\mathbf{v}_0)$ is the maximal value of q_A on S, we know that $q_A(\mathbf{x}) \leq \lambda_A$ whenever $\|\mathbf{x}\| = 1$. This implies that for *every* $\mathbf{v} \in \mathbf{R}^n$, we have $q_A(\mathbf{v}) \leq \lambda_A\|\mathbf{v}\|^2$. Indeed, this inequality is clear when $\mathbf{v} = \mathbf{0}$ (as $q_A(\mathbf{0}) = 0$), and if $\mathbf{v} \neq \mathbf{0}$ then we can form the unit vector $\mathbf{x} = \mathbf{v}/\|\mathbf{v}\|$ to obtain $q_A(\mathbf{v}/\|\mathbf{v}\|) \leq \lambda_A$. But $q_A(c\mathbf{y}) = c^2q_A(\mathbf{y})$ for any scalar c and vector $\mathbf{y}$, so setting $c = 1/\|\mathbf{v}\|$ and $\mathbf{y} = \mathbf{v}$ then gives $q_A(\mathbf{v}/\|\mathbf{v}\|) = q_A(\mathbf{v})/\|\mathbf{v}\|^2$. Hence, $q_A(\mathbf{v})/\|\mathbf{v}\|^2 \leq \lambda_A$, so $q_A(\mathbf{v}) \leq \lambda_A\|\mathbf{v}\|^2$ as desired.

Our goal is to show $A\mathbf{v}_0 = \lambda_A\mathbf{v}_0$. To this end, we are going to introduce auxiliary polynomials that control the possible failure of this equality. Fix $\mathbf{h} \in \mathbf{R}^n$ and define $f_{\mathbf{h}}(t) = \lambda_A\|\mathbf{v}_0 + t\mathbf{h}\|^2 - q_A(\mathbf{v}_0 + t\mathbf{h})$. Note that $f_{\mathbf{h}}(t) \geq 0$ for any t (since we have shown that $q_A(\mathbf{v}) \leq \lambda_A\|\mathbf{v}\|^2$ for every $\mathbf{v} \in \mathbf{R}^n$). The function $f_{\mathbf{h}}(t)$ is a polynomial in t of a special type:

$$\begin{aligned}
f_{\mathbf{h}}(t) &= \lambda_A(\mathbf{v}_0 + t\mathbf{h}) \cdot (\mathbf{v}_0 + t\mathbf{h}) - (\mathbf{v}_0 + t\mathbf{h}) \cdot A(\mathbf{v}_0 + t\mathbf{h}) \\
&= \lambda_A(\mathbf{v}_0 + t\mathbf{h}) \cdot (\mathbf{v}_0 + t\mathbf{h}) - (\mathbf{v}_0 + t\mathbf{h}) \cdot (A\mathbf{v}_0 + tA\mathbf{h}) \\
&= \lambda_A(\mathbf{v}_0 \cdot \mathbf{v}_0 + 2t\mathbf{v}_0 \cdot \mathbf{h} + t^2\mathbf{h} \cdot \mathbf{h}) - (\mathbf{v}_0 \cdot A\mathbf{v}_0 + t\mathbf{v}_0 \cdot A\mathbf{h} + t\mathbf{h} \cdot A\mathbf{v}_0 + t^2\mathbf{h} \cdot A\mathbf{h}) \\
&= \lambda_A(\mathbf{v}_0 \cdot \mathbf{v}_0) + 2\lambda_A t\mathbf{v}_0 \cdot \mathbf{h} + \lambda_A(\mathbf{h} \cdot \mathbf{h})t^2 - q_A(\mathbf{v}_0) - tA\mathbf{v}_0 \cdot \mathbf{h} - t\mathbf{h} \cdot A\mathbf{v}_0 - (\mathbf{h} \cdot A\mathbf{h})t^2 \\
&= \lambda_A + 2\lambda_A(\mathbf{v}_0 \cdot \mathbf{h})t + \lambda_A\|\mathbf{h}\|t^2 - \lambda_A - 2(A\mathbf{v}_0 \cdot \mathbf{h})t - q_A(\mathbf{h})t^2 \\
&= 2\lambda_A(\mathbf{v}_0 \cdot \mathbf{h})t - 2(A\mathbf{v}_0 \cdot \mathbf{h})t + \lambda_A\|\mathbf{h}\|^2t^2 - q_A(\mathbf{h})t^2 \\
&= 2((\lambda_A\mathbf{v}_0) \cdot \mathbf{h})t - 2(A\mathbf{v}_0 \cdot \mathbf{h})t + (\lambda_A\|\mathbf{h}\|^2 - q_A(\mathbf{h}))t^2 \\
&= 2((\lambda_A\mathbf{v}_0 - A\mathbf{v}_0) \cdot \mathbf{h})t + (\lambda_A\|\mathbf{h}\|^2 - q_A(\mathbf{h}))t^2.
\end{aligned}$$

In other words, $f_{\mathbf{h}}(t) = at + bt^2$ where $a = 2((\lambda_A\mathbf{v}_0 - A\mathbf{v}_0) \cdot \mathbf{h})$ and $b = \lambda_A\|\mathbf{h}\|^2 - q_A(\mathbf{h})$. Note that $b \geq 0$ (again using that $q_A(\mathbf{v}) \leq \lambda_A\|\mathbf{v}\|^2$ for every $\mathbf{v} \in \mathbf{R}^n$).

We are going to show $a = 0$. For every t, we have $0 \leq f_{\mathbf{h}}(t) = at + bt^2 = t(a + bt)$. If $b = 0$ then $0 \leq f_{\mathbf{h}}(t) = at$ for all t, so setting $t = -a$ yields $0 \leq -(a^2)$, which forces $a = 0$. Suppose instead that $b \neq 0$ (so $b > 0$). The function $f_{\mathbf{h}}$ is then a quadratic polynomial with positive leading coefficient, so its graph is a parabola with minimal value attained at $t = -a/(2b)$. But that minimal value is $-(a/(2b))(a - a/2) = -a^2/(4b) \leq 0$, yet $f_{\mathbf{h}}(t) \geq 0$ for all t, so once again we conclude that $a = 0$.

The upshot is that $2(\lambda_A\mathbf{v}_0 - A\mathbf{v}_0) \cdot \mathbf{h} = 0$ no matter what $\mathbf{h} \in \mathbf{R}^n$ we choose. Taking $\mathbf{h}$ to be $\lambda_A\mathbf{v}_0 - A\mathbf{v}_0$, we get that $2\|\lambda_A\mathbf{v}_0 - A\mathbf{v}_0\|^2 = 0$, so $\lambda_A\mathbf{v}_0 - A\mathbf{v}_0 = \mathbf{0}$. Hence, $A\mathbf{v}_0 = \lambda_A\mathbf{v}_0$ as desired.

B.4. A geometric explanation of SVD. In Section 27.4 we used the Spectral Theorem to sketch a proof of the main result about SVD (Theorem 27.3.3) via matrix algebra. Here we provide a more complete and totally different proof of that main result. The argument will be rather longer than the proof via matrix algebra, but the length of the proof is not really an indication of being more difficult: it is entirely due to the fact that certain concepts in the argument (such as quadratic form, dot product, and linear function) have been introduced in this book in a manner that is not sufficiently "coordinate-free". If we had set up the framework of linear algebra in this book in a more general way (as is done in Math 113) then the geometric argument below would be a lot shorter.

PROOF. Consider the linear transformation $L : \mathbf{R}^n \to \mathbf{R}^m$ with matrix A (that is: $L(\mathbf{x}) = A\mathbf{x}$ for all $\mathbf{x} \in \mathbf{R}^n$). Let's call its null space $V = N(A) \subset \mathbf{R}^n$ and its column space $W = C(A) \subset \mathbf{R}^m$. We are going to carry out some geometric arguments involving W and the orthogonal subspace $V^\perp \subset \mathbf{R}^n$ consisting of vectors in $\mathbf{R}^n$ perpendicular to everything in V.

Step 1. By Theorem 6.2.4, every vector $\mathbf{x} \in \mathbf{R}^n$ is uniquely of the form $\mathbf{v} + \mathbf{v}'$ with $\mathbf{v}$ belong to the null space V and $\mathbf{v}'$ belong to its orthogonal subspace $V^\perp$. In the language of projections onto subspaces, $\mathbf{v} = \mathbf{Proj}_V(\mathbf{x})$ and

$$\mathbf{v}' = \mathbf{Proj}_{V^\perp}(\mathbf{x}) = \mathbf{x} - \mathbf{Proj}_V(\mathbf{x}).$$

Since V is the null space of A and $\mathbf{v} \in V$, we have $L(\mathbf{v}) = A\mathbf{v} = \mathbf{0}$. Thus,

$$L(\mathbf{x}) = L(\mathbf{v} + \mathbf{v}') = L(\mathbf{v}) + L(\mathbf{v}') = \mathbf{0} + L(\mathbf{v}') = L(\mathbf{v}'),$$

so the effect of L on $\mathbf{x}$ only matters through what it does to $V^\perp$. Hence, everything in the column space of A (i.e., any vector of the form $L(\mathbf{x}) = A\mathbf{x}$) has the form $L(\mathbf{v}')$ for some $\mathbf{v}' \in V^\perp$.

The vectors of the form $L(\mathbf{x})$ are exactly those in the column space $W \subset \mathbf{R}^m$ of L (this is exactly Theorem 13.4.1), and we have just seen that every $\mathbf{w} \in W$ has the form $L(\mathbf{v}')$ for some $\mathbf{v}' \in V^\perp \subset \mathbf{R}^n$. We claim that $\mathbf{w}$ arises in this way from *exactly one* $\mathbf{v}' \in V^\perp$; that is, if

$$L(\mathbf{v}_1') = \mathbf{w} = L(\mathbf{v}_2')$$

for some $\mathbf{v}_1', \mathbf{v}_2' \in V^\perp$ then we claim $\mathbf{v}_1' = \mathbf{v}_2'$. This says that $L : V^\perp \to W$ is a linear "one-to-one correspondence": everything in W has the form $L(\mathbf{v}')$ for exactly one $\mathbf{v}' \in V^\perp$.

To prove $\mathbf{v}_1' = \mathbf{v}_2'$, we show the difference $\mathbf{v}_1' - \mathbf{v}_2'$ vanishes. This difference belongs to $V^\perp$ since $V^\perp$ is a linear subspace and any difference of vectors in a subspace is still in that subspace. Next,

$$A(\mathbf{v}_1' - \mathbf{v}_2') = L(\mathbf{v}_1' - \mathbf{v}_2') = L(\mathbf{v}_1') - L(\mathbf{v}_2') = \mathbf{w} - \mathbf{w} = \mathbf{0},$$

forcing the difference vector $\mathbf{v}_1' - \mathbf{v}_2' \in V^\perp$ to also belong to the null space V of A. But a vector belonging to both V *and* $V^\perp$ has to vanish: the orthogonality of any $\mathbf{y} \in V^\perp$ to *everything* in V (such as to $\mathbf{y}$ if $\mathbf{y} \in V$ too) implies that if $\mathbf{y} \in V$ also then $0 = \mathbf{y} \cdot \mathbf{y} = \|\mathbf{y}\|^2$, so $\mathbf{y} = \mathbf{0}$. Using $\mathbf{y} = \mathbf{v}_1' - \mathbf{v}_2'$ implies $\mathbf{v}_1' - \mathbf{v}_2' = \mathbf{0}$, so $\mathbf{v}_1' = \mathbf{v}_2'$ as desired.

Step 2. Now comes the great geometric idea, with which we will be able to use the Spectral Theorem: we will use the linear transformation $L : V^\perp \to W$ to transport the dot product on $W \subset \mathbf{R}^m$ to an operation on the subspace $V^\perp \subset \mathbf{R}^n$ that *behaves* like a dot product but algebraically looks nothing at all like the *usual* dot product from the viewpoint of coordinates on $\mathbf{R}^n$. Define

$$\mathbf{v}_1' * \mathbf{v}_2' = L(\mathbf{v}_1') \cdot L(\mathbf{v}_2')$$

for any $\mathbf{v}_1', \mathbf{v}_2' \in V^\perp$. This may look a bit mysterious, but we claim it behaves *exactly* like the usual dot product in the sense that (i) it shares all of the usual algebraic properties with respect to vector addition and scalar multiplication in $V^\perp$, and (ii) it satisfies the "positive-definiteness" property

$$\mathbf{v}' * \mathbf{v}' \geq 0, \text{ with equality precisely when } \mathbf{v}' = \mathbf{0}.$$

Roughly speaking, we are using $L : V^\perp \to W$ to transport the "geometry of $\mathbf{R}^m$ seen in W" over into the setting of the subspace $V^\perp \subset \mathbf{R}^n$. This makes L serve as a dictionary between different worlds.

To understand what is going on with $*$, first note that for the algebraic features of dot products the *linearity* of L does all of the work. For instance:

$$\begin{aligned}(\mathbf{v}_1' + \mathbf{v}_2') * \mathbf{v}_3' = L(\mathbf{v}_1' + \mathbf{v}_2') \cdot L(\mathbf{v}_3') &= (L(\mathbf{v}_1') + L(\mathbf{v}_2')) \cdot L(\mathbf{v}_3') \\ &= L(\mathbf{v}_1') \cdot L(\mathbf{v}_3') + L(\mathbf{v}_2') \cdot L(\mathbf{v}_3') \\ &= \mathbf{v}_1' * \mathbf{v}_3' + \mathbf{v}_2' * \mathbf{v}_3'.\end{aligned}$$

The other algebraic properties of dot products (as listed in parts (i), (iii), and (iii′) of Theorem 2.2.1) carry over in exactly the same way. To show $\mathbf{v}' * \mathbf{v}' \geq 0$ with equality precisely when $\mathbf{v}' = \mathbf{0}$, we observe that $\mathbf{v}' * \mathbf{v}' = L(\mathbf{v}') \cdot L(\mathbf{v}') = \|L(\mathbf{v}')\|^2 \geq 0$ with equality precisely when $L(\mathbf{v}') = \mathbf{0}$, which is exactly when $\mathbf{v}' = \mathbf{0}$ by the uniqueness property of $L : V^{\perp} \to W$ at the end of Step 1.

With this motivation, we are led to consider the function $q : V^{\perp} \to \mathbf{R}$ defined by

$$q(\mathbf{v}') = \mathbf{v}' * \mathbf{v}' = L(\mathbf{v}') \cdot L(\mathbf{v}') = \|L(\mathbf{v}')\|^2,$$

regarded as a kind of shadow of squared-length on W seen from the perspective of $V^{\perp}$; in other words, we use L to define a non-standard notion of "squared length" on $V^{\perp} \subset \mathbf{R}^n$ by using squared length on $L(V^{\perp}) = W$.

We claim that q really looks like a quadratic form when written in terms of coordinates relative to a basis of $V^{\perp}$. Indeed, since any basis of $V^{\perp}$ can be extended to a basis of $\mathbf{R}^n$ by combining it with a basis of V, it suffices to show the same "quadratic form" property for the function $L(\mathbf{x}) \cdot L(\mathbf{x})$ on the entirety of $\mathbf{R}^n$ (then we can set to 0 the coordinates in that expression corresponding to basis vectors from V). But in the language of matrix-vector products,

$$L(\mathbf{x}) \cdot L(\mathbf{x}) = (A\mathbf{x}) \cdot (A\mathbf{x}) = (A\mathbf{x})^{\top}(A\mathbf{x}) = \mathbf{x}^{\top}(A^{\top}A)\mathbf{x}$$

(the second equality uses (20.1.1)); this is the quadratic form corresponding to the symmetric matrix $M = A^{\top}A$ (as discussed in Section 20.3). If we write it in terms of some basis of $\mathbf{R}^n$ that may not be the standard one then it still looks like a "quadratic expression" in the vector coordinates (since expressing n-vectors in terms of a non-standard basis involves coefficients that are linear expressions in the usual vector entries, and plugging such linear expressions into a quadratic form again yields a quadratic form). Thus, the function $q : V^{\perp} \to \mathbf{R}$ really does look algebraically like a quadratic form when vectors in $V^{\perp}$ are expressed in terms of a fixed choice of basis, and we have seen that this function on $V^{\perp}$ is even "positive-definite": for $\mathbf{x} \in V^{\perp}$, $q(\mathbf{x}) = \|L\mathbf{x}\|^2 \geq 0$ with equality precisely when $\mathbf{x} = \mathbf{0}$.

Step 3. Finally, we can apply the consequence (24.2.2) of the Spectral Theorem to describe the "positive-definite quadratic form" $q : V^{\perp} \to \mathbf{R}$. Strictly speaking, (24.2.2) is for quadratic forms on Euclidean spaces $\mathbf{R}^N$ equipped with their usual dot product, whereas q is given on $V^{\perp}$. Hence, next we make $V^{\perp}$ "look like" a Euclidean space. Let $d = \dim V^{\perp}$ and pick an orthonormal basis $B' = \{\mathbf{e}'_1, \ldots, \mathbf{e}'_d\}$ of $V^{\perp}$ (Theorem 5.2.5), so the *usual* dot product of $\mathbf{R}^n$ viewed on $V^{\perp}$ looks like the *usual* dot product of $\mathbf{R}^d$ when written in terms of the B'-coordinates: $(\sum_{j=1}^{d} x_j \mathbf{e}'_j) \cdot (\sum_{i=1}^{d} y_i \mathbf{e}'_i) = \sum_{1 \leq i,j \leq d} x_j y_i \mathbf{e}'_j \cdot \mathbf{e}'_i = \sum_{j=1}^{d} x_j y_j$, where the final equality holds because $\mathbf{e}'_j \cdot \mathbf{e}'_i = 0$ when $j \neq i$ and $\mathbf{e}'_j \cdot \mathbf{e}'_j = 1$. The point is that $\sum_{j=1}^{d} x_j y_j$ is the formula for the usual dot product on $\mathbf{R}^d$, but it is now expressed in terms of the B'-coordinates on $V^{\perp}$. So by writing everything in terms of such an orthonormal basis we can make $V^{\perp}$ equipped with the dot product it inherits from $\mathbf{R}^n$ *look like* the Euclidean space $\mathbf{R}^d$ equipped with its usual dot product.

In Step 2 we saw that when $q : V^{\perp} \to \mathbf{R}$ is written in terms of *any* basis of $V^{\perp}$ then it looks like some (mysterious) quadratic form in d variables, so for some (mysterious) coefficients c_{ij} we have $q(\sum_{j=1}^{d} x_j \mathbf{e}'_j) = \sum_{1<i<j\leq d} c_{ij} x_i x_j$ that we moreover know is positive-definite. By (24.2.2), the Spectral Theorem provides an orthonormal basis $\{\mathbf{v}'_1, \ldots, \mathbf{v}'_d\}$ of $\mathbf{R}^d$ for which the expression on the right side becomes "diagonal" (no cross-terms) when we write everything in $\mathbf{R}^d$ in terms of the orthonormal basis $\{\mathbf{v}'_1, \ldots, \mathbf{v}'_d\}$ of $\mathbf{R}^d$ (rather than in terms of the standard basis).

By using B'-coordinates to identify points of $\mathbf{R}^d$ with points of $V^{\perp}$, which we have seen *respects the notions of dot product on both sides* (and so respects the notions of orthogonality and of unit vector), we can regard $\{\mathbf{v}'_1, \ldots, \mathbf{v}'_d\}$ as an *orthonormal basis of* $V^{\perp}$ for which q looks "diagonal":

$$q(\sum y_j \mathbf{v}'_j) = \sum \lambda_j y_j^2 \tag{B.4.1}$$

for some $\lambda_1, \dots, \lambda_d$. In particular $q(\mathbf{v}'_j) = \lambda_j$ and q is positive-definite, so $\lambda_j > 0$ for all j. By rearranging the $\mathbf{v}'_j$'s if necessary, we can ensure $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d > 0$.

In Step 4 we are going to explain the geometric meaning of (B.4.1), but for motivational purposes it is instructive to show the picture of that visualization right now. It will turn out that when we describe the linear transformation $L : V^\perp \to W$ in terms of the orthonormal basis $\{\mathbf{v}'_1, \dots, \mathbf{v}'_d\}$ of $V^\perp$ and the basis $\{\mathbf{w}'_1 = L(\mathbf{v}'_1)/\|L(\mathbf{v}'_1)\|, \dots, \mathbf{w}'_d = L(\mathbf{v}'_d)/\|L(\mathbf{v}'_d)\|\}$ of unit vectors for W then L carries the "ellipsoid" $\sum_{j=1}^{d} \lambda_j y_j^2 = 1$ in $V^\perp$ whose axes of symmetry have lengths $1/\sqrt{\lambda_1} \leq 1/\sqrt{\lambda_2} \leq \cdots \leq 1/\sqrt{\lambda_d}$ onto the unit sphere $\sum_{j=1}^{d} z_j^2 = 1$ in W, as shown in Figure B.4.1 with $\mathbf{v}_j = \mathbf{v}'_j/\sqrt{\lambda_j}$ (corresponding to a case with $V = N(A) = \{\mathbf{0}\}$ and $W = \mathbf{R}^d = \mathbf{R}^3$, so $V^\perp = \mathbf{R}^d = \mathbf{R}^3$).

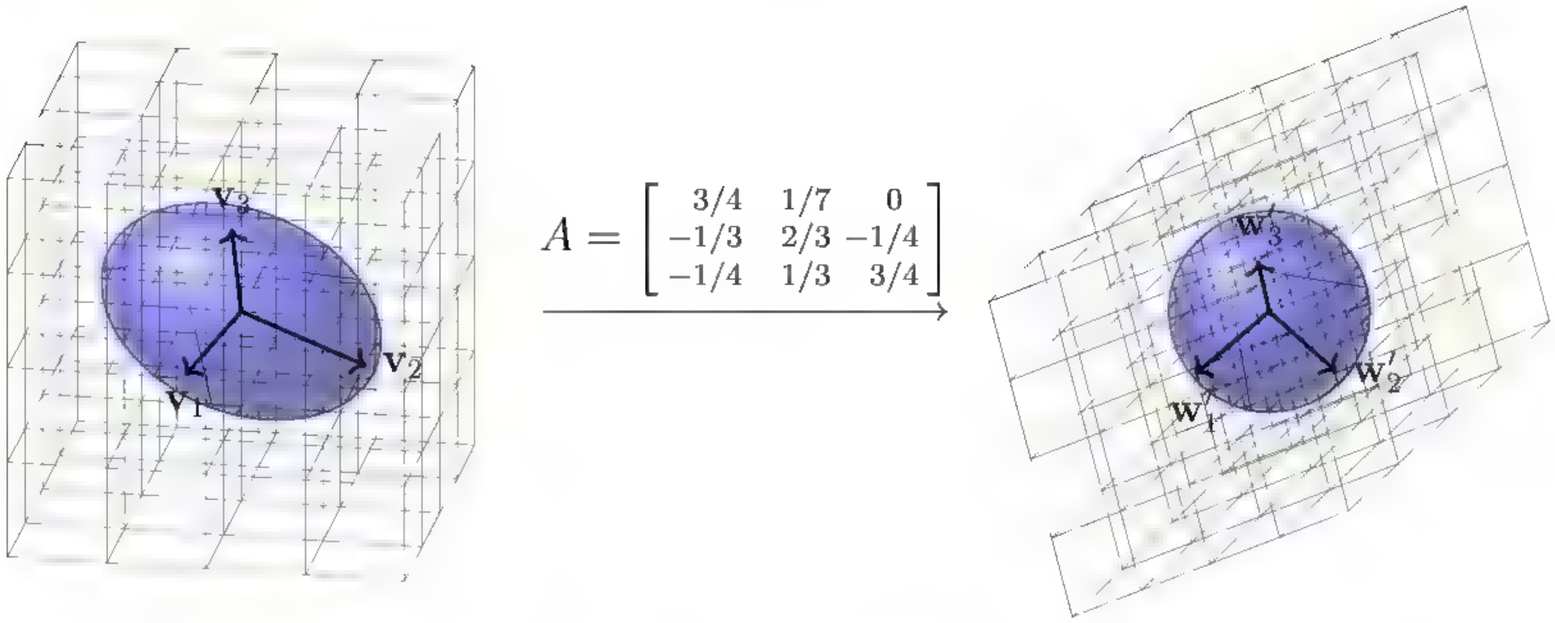

$\mathbf{v}_3$
$\mathbf{v}_2$
$\mathbf{v}_1$
$A = \begin{bmatrix} 3/4 & 1/7 & 0 \\ -1/3 & 2/3 & -1/4 \\ -1/4 & 1/3 & 3/4 \end{bmatrix}$
$\mathbf{w}'_3$
$\mathbf{w}'_2$
$\mathbf{w}'_1$

FIGURE B.4.1. Proof of the singular value decomposition for a specific invertible 3×3 matrix A defining some $L : \mathbf{R}^3 \to \mathbf{R}^3$, with $\|\mathbf{v}_1\| \approx 1.03789$, $\|\mathbf{v}_2\| \approx 1.50541$, $\|\mathbf{v}_3\| \approx 1.32746$. Though A highly distorts the standard grid, it carries *some* ellipsoid onto a sphere!

In accordance with the principle that a picture is worth a thousand words, Figure B.4.1 expresses the entire content of the geometric proof of SVD. Before we carry out Step 4, let's explain what Figure B.4.1 is conveying and why it is so surprising. The corners of the cubical grid in the source are the points in $\mathbf{R}^3$ with integer coordinates, and the parallelepiped grid in the target is where the cubical grid is carried under L; the corners of the smallest parallelepipeds are the linear combinations $a_1 L(\mathbf{e}_1) + a_2 L(\mathbf{e}_2) + a_3 L(\mathbf{e}_3) = L(a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + a_3\mathbf{e}_3)$ with integers a_1, a_2, a_3 ($L(\mathbf{e}_i)$ is the ith column of A). The deviation of the grid in the target from the cubical grid along the coordinate axes thereby illustrates the stretching, shearing, and rotation effects typical of linear transformations. This deviation from the cubical coordinate grid, especially the fact that the angles at the corners of the basic building blocks are not right angles (unlike for the cubical coordinate grid that it arises from in the source), suggests that it would be very rare for there to exist 3 mutually orthogonal directions in the source that are carried by L onto 3 mutually orthogonal directions in the target.

The main content of SVD is that such magical directions in the source $V^\perp$ and target W *always* exist! We can't see them in an elementary way because when staring at the matrix A we are blinded by the lack of nice geometry among the columns (due to the non-cubical nature of the grid in the target and especially its typical lack of right angles). In effect, the "usual coordinates" for describing vectors and matrices may be poorly suited to seeing good geometric behavior of L. Step 4 will show that $\|L(\mathbf{v}'_j)\| = \sqrt{\lambda_j}$ (so $L(\mathbf{v}_j) = \mathbf{w}'_j$ for $\mathbf{v}_j = \mathbf{v}'_j/\sqrt{\lambda_j}$ with $\|\mathbf{v}_j\| = 1/\sqrt{\lambda_j}$) and that the unit vectors $\mathbf{w}'_1, \dots, \mathbf{w}'_d$ are mutually orthogonal, so the mutually orthogonal (typically non-unit) vectors $\mathbf{v}_j$ in the source are respectively carried onto the mutually orthogonal unit vectors $\mathbf{w}'_j$ in the target. These outcomes of Step 4, which are the heart of SVD for A, are expressed visually in two ways: (i) the ellipsoid in the source with semi-axes given by the $\mathbf{v}_j$'s is carried onto the unit sphere in the target, (ii) the axes of symmetry of the ellipsoid are carried onto *mutually perpendicular* directions in the target (through the $\mathbf{w}'_j$'s).

Step 4. Now we show that the diagonalization (B.4.1) of q expresses the singular value decomposition. Just as the usual dot product is recovered from squared-length via the formula (20.4.2) which we used just after Theorem 20.4.1 to show that a length-preserving linear transformation is also angle-preserving, we have an analogous formula recovering $*$ from q:

$$\mathbf{x}' * \mathbf{y}' = \frac{(\mathbf{x}' + \mathbf{y}') * (\mathbf{x}' + \mathbf{y}') - \mathbf{x}' * \mathbf{x}' - \mathbf{y}' * \mathbf{y}'}{2} = \frac{q(\mathbf{x}' + \mathbf{y}') - q(\mathbf{x}') - q(\mathbf{y}')}{2}.$$

Indeed, this holds for exactly the same reason as (20.4.2): $*$ shares the same relevant algebraic properties as the usual dot product! Hence, for $i \neq j$ we have

$$\mathbf{v}'_j * \mathbf{v}'_i = \frac{q(\mathbf{v}'_j + \mathbf{v}'_i) - q(\mathbf{v}'_j) - q(\mathbf{v}'_i)}{2} = \frac{(\lambda_j + \lambda_i) - \lambda_j - \lambda_i}{2} = 0.$$

This says that $L(\mathbf{v}'_j) \cdot L(\mathbf{v}'_i) = 0$ for $i \neq j$, so $\{L(\mathbf{v}'_1), \dots, L(\mathbf{v}'_d)\}$ is an *orthogonal* set of vectors in W. Provided it is a spanning set consisting of nonzero vectors, we could conclude that it is an orthogonal basis of W.

Each $\mathbf{w} \in W$ has the form $L(\mathbf{v}')$ for some $\mathbf{v}' \in V^{\perp}$, so by writing $\mathbf{v}' = \sum_{j=1}^d c_j \mathbf{v}'_j$ we have

$$\mathbf{w} = L(\mathbf{v}') = \sum_{j-1}^{d} c_j L(\mathbf{v}'_j).$$

Hence, the vectors $L(\mathbf{v}'_1), \dots, L(\mathbf{v}'_d)$ do span W. Moreover, each $L(\mathbf{v}'_j)$ is nonzero since more generally $L(\mathbf{v}')$ is nonzero for any nonzero $\mathbf{v}' \in V^{\perp}$. Indeed, we know that everything in W has the form $L(\mathbf{x})$ for a unique $\mathbf{x} \in V^{\perp}$, so by uniqueness the only way we can have $L(\mathbf{x}) = \mathbf{0}$ with $\mathbf{x} \in V^{\perp}$ is for $\mathbf{x} = \mathbf{0}$. Hence, for $\mathbf{v}' \in V^{\perp}$ that is nonzero, definitely $L(\mathbf{v}')$ is nonzero. This concludes the verification that $L(\mathbf{v}'_1), \dots, L(\mathbf{v}'_d)$ is an orthogonal basis of W.

Of course, the $L(\mathbf{v}'_j)$'s are typically not unit vectors: $\|L(\mathbf{v}'_j)\|^2 = L(\mathbf{v}'_j) \cdot L(\mathbf{v}'_j) = q(\mathbf{v}_j) = \lambda_j$. Letting $\sigma_j = \sqrt{\lambda_j} = \|L(\mathbf{v}'_j)\| > 0$, we have $\sigma_1 \geq \cdots \geq \sigma_d > 0$ and the vectors

$$\mathbf{w}'_j = \frac{L(\mathbf{v}'_j)}{\sigma_j}$$

are unit vectors which constitute an *orthonormal basis* of W. To summarize, we have found orthonormal bases $\{\mathbf{v}'_1, \dots, \mathbf{v}'_d\}$ of $V^{\perp}$ and $\{\mathbf{w}'_1, \dots, \mathbf{w}'_d\}$ of W for which $L(\mathbf{v}'_j) = \sigma_j \mathbf{w}'_j$ for all $1 \leq j \leq d$. This looks very much like an SVD, except we are working with orthonormal bases of $V^{\perp}$ and W rather than of the ambient spaces $\mathbf{R}^n$ and $\mathbf{R}^m$ that contain them. The final Step 5 below disposes of this minor issue.

Remark B.4.1. Now that Step 4 is complete, before carrying out Step 5 let's pause to revisit Figure B.4.1 to get more insight into the geometry of what we just did. Consider the collection E of points $\mathbf{w} = \sum_{j=1}^d z_j \mathbf{w}'_j \in W$ satisfying $\sum_{j=1}^d a_j z_j^2 = 1$ for some coefficients $a_j > 0$ (E is an "ellipsoid"). For which $\mathbf{v}' = \sum_{j=1}^d y_j \mathbf{v}'_j \in V^{\perp}$ do we have $L(\mathbf{v}') \in E$? Since $L(\sum_{j=1}^d y_j \mathbf{v}'_j) = \sum_{j=1}^d y_j L(\mathbf{v}'_j) = \sum_{j=1}^d y_j \sigma_j \mathbf{w}'_j$, the condition $L(\mathbf{v}') \in E$ says exactly that $1 = \sum_{j-1}^d a_j (y_j \sigma_j)^2 = \sum_{j-1}^d a_j \sigma_j^2 y_j^2 = \sum_{j-1}^d a_j \lambda_j y_j^2$. When describing vectors in $V^{\perp}$ in terms of the orthonormal basis of $\mathbf{v}'_j$'s and describing vectors in W in terms of the orthonormal basis of $\mathbf{w}'_j$'s, there are two notable special cases of this conclusion:

(i) Using $a_j = 1$ for all j, L carries the ellipsoid $\sum_{j=1}^d \lambda_j y_j^2 = 1$ in $V^{\perp}$ onto the unit sphere in W.
(ii) Using $a_j = 1/\lambda_j$ for all j, L carries the unit sphere in $V^{\perp}$ onto the ellipsoid $\sum_{j-1}^d z_j^2/\lambda_j = 1$ in W.

Case (ii) ("unit sphere onto ellipsoid") is what is shown in visualizations of SVD in all other references (books, Internet, etc.). But case (i) ("ellipsoid onto unit sphere") is what is shown in Figure B.4.1, and this is really the correct picture for illustrating the geometric proof of SVD. To see this, note that the entire content of the preceding steps has been to build orthonormal bases $\{\mathbf{v}'_1, \dots, \mathbf{v}'_d\}$ of $V^\perp$ and $\{\mathbf{w}'_1, \dots, \mathbf{w}'_d\}$ of W so that composing with $L : V^\perp \to W$ carries the positive-definite squared-length quadratic form on W back to a positive-definite "diagonal" quadratic form q on $V^\perp$. Such a matching of positive-definite quadratic forms under composing with L is expressed most naturally by saying composing with L matches each level set for squared-length on W with the level set (for the same value) for a specific diagonal quadratic form q on $V^\perp$. For any positive-definite quadratic form Q and $c > 0$, the level set $Q = c$ is obtained from the level set $Q = 1$ via multiplication by $\sqrt{c}$ (since $Q(\sqrt{c}\,\mathbf{x}) = (\sqrt{c})^2 Q(\mathbf{x}) = cQ(\mathbf{x})$), so matching level sets of such quadratic forms is encoded entirely by matching the level sets for value 1. That in turn is *exactly* what is shown in Figure B.4.1 (ellipsoid onto unit sphere).

Step 5. If we append an orthonormal basis of V to the end of the list of $\mathbf{v}'_j$'s then by Theorem 6.2.4 the extended list is an orthonormal basis $\mathcal{B}'$ of $\mathbf{R}^n$; by design its last $n - d$ members are annihilated by L (because V is the null space of A). Likewise, we can extend $\{\mathbf{w}'_1, \dots, \mathbf{w}'_d\}$ to an orthonormal basis $\mathcal{B}$ of $\mathbf{R}^m$ by appending to the end of that list an orthonormal basis of $W^\perp$. When the effect of the linear transformation $L : \mathbf{R}^n \to \mathbf{R}^m$ is expressed by writing vectors in $\mathbf{R}^n$ and $\mathbf{R}^m$ respectively in terms of these chosen orthonormal bases of $\mathbf{R}^n$ and $\mathbf{R}^m$ then it is exactly an $m \times n$ "diagonal" matrix D with diagonal entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$ by defining $\sigma_i = 0$ for all $i > d$.

Let Q be the $m \times m$ matrix whose columns are the members $\mathbf{w}'_1, \dots, \mathbf{w}'_d, \dots$ of the orthonormal basis $\mathcal{B}$ of $\mathbf{R}^m$, and let Q' be the $n \times n$ matrix whose columns are the members $\mathbf{v}'_1, \dots, \mathbf{v}'_d, \dots$ of the orthonormal basis $\mathcal{B}'$ of $\mathbf{R}^n$. Each of Q and Q' are orthogonal matrices. Moreover, essentially by design, the effect of Q is to turn $\mathcal{B}$-coordinates on $\mathbf{R}^m$ into "standard coordinates" on $\mathbf{R}^m$, and the effect of Q' is to turn $\mathcal{B}'$-coordinates on $\mathbf{R}^n$ into "standard coordinates" on $\mathbf{R}^n$. Hence, going in reverse, the effect of $Q'^{-1} = Q'^\top$ is to turn "standard coordinates" on $\mathbf{R}^n$ into $\mathcal{B}'$-coordinates on $\mathbf{R}^n$.

Putting it all together, the effect of the $m \times n$ matrix $QDQ'^\top$ is first to turn standard coordinates on $\mathbf{R}^n$ into $\mathcal{B}'$-coordinates on $\mathbf{R}^n$, then apply the "diagonal" D, and finally turn that output viewed in $\mathcal{B}$-coordinates on $\mathbf{R}^m$ back into standard coordinates on $\mathbf{R}^m$. In other words, the linear transformation $\mathbf{R}^n \to \mathbf{R}^m$ associated with the $m \times n$ matrix $QDQ'^\top$ is the one that looks like D when we write vectors $\mathbf{R}^n$ in terms of $\mathcal{B}'$-coordinates and we write vectors in $\mathbf{R}^m$ in terms of $\mathcal{B}$-coordinates. But we have designed the bases $\mathcal{B}$ and $\mathcal{B}'$ so that $L : \mathbf{R}^n \to \mathbf{R}^m$ is *precisely* that same overall linear transformation, yet L corresponds to the $m \times n$ matrix A by definition. Since a linear transformation $\mathbf{R}^n \to \mathbf{R}^m$ corresponds to *exactly one* $m \times n$ matrix, we conclude that

$$QDQ'^\top = A.$$

Voila, we have established the singular value decomposition for A!

If one inspects the geometric meaning of the construction in Steps 1–4, one sees that an SVD for A is exactly such a geometric construction in disguise. In particular, the nonzero σ_j's are uniquely determined because they must be exactly the square roots of the positive eigenvalues of the "symmetric matrix" corresponding to the "quadratic form" $q : V^\perp \to \mathbf{R}$ as built above (when we work with any choice of orthonormal basis of $V^\perp$ to make $V^\perp$ look like a Euclidean space $\mathbf{R}^d$). ☐

We now use SVD to prove the Polar Decomposition stated and used in Example 24.6.7:

Theorem B.4.2 (Polar Decomposition). If A is an invertible $n \times n$ matrix then we can uniquely write $A = QS$ where Q is an orthogonal $n \times n$ matrix and S is a positive-definite symmetric $n \times n$ matrix. Moreover, also $A = S'Q$ for the same Q and a positive-definite symmetric $n \times n$ matrix S'.

There is a vast generalization of this theorem called the Cartan[28] Decomposition which is important in the study of "continuous symmetry".

PROOF. Once we prove the assertions concerning Q and S, we can obtain the expression $S'Q$ with the same Q as follows. Since $A = QS = QS(Q^{-1}Q) = (QSQ^{-1})Q = (QSQ^\top)Q$, we can define $S' = QSQ^\top$ provided that this S' really is symmetric and positive-definite. For its symmetry, we compute

$$(S')^\top = (QSQ^\top)^\top = (Q^\top)^\top S^\top Q^\top = QSQ^\top = S'$$

(the second equality uses that transpose flips the order of matrix multiplication). For positive-definiteness, for any nonzero $\mathbf{x} \in \mathbf{R}^n$ we want to show that $q_{S'}(\mathbf{x}) = \mathbf{x} \cdot S'\mathbf{x}$ is positive. Since $Q^\top\mathbf{x} \neq \mathbf{0}$ (by invertibility of $Q^\top$), we have

$$\mathbf{x} \cdot S'\mathbf{x} = \mathbf{x} \cdot (QSQ^\top)\mathbf{x} = \mathbf{x} \cdot Q(SQ^\top\mathbf{x}) = (Q^\top\mathbf{x}) \cdot (SQ^\top\mathbf{x}) = q_S(Q^\top\mathbf{x}) > 0$$

by positive-definiteness of S (the third equality uses Theorem 20.1.4). Hence, we now may focus our attention on the assertions involving Q and S. For the existence of the desired expression QS for A, we shall manipulate the singular value decomposition for A (so the Spectral Theorem is being used).

Step 1. The SVD for A says that $A = Q_0D_0{Q_0'}^\top = Q_0D_0{Q_0'}^{-1}$ for some $n \times n$ orthogonal matrices Q_0 and Q_0' and diagonal matrix D_0 with non-negative diagonal entries. The matrix A is invertible by hypothesis, and Q_0 and Q_0' are invertible since they are square orthogonal matrices, so $D_0 = Q_0^{-1}AQ_0'$ is invertible (as is any product of invertible $n \times n$ matrices). An invertible diagonal matrix must have all diagonal entries nonzero, so the non-negative diagonal entries of D_0 are all positive. In particular, the visibly symmetric D_0 is also positive-definite. Writing

$$A = Q_0D_0{Q_0'}^{-1} = Q_0({Q_0'}^{-1}Q_0')D_0{Q_0'}^{-1} = (Q_0{Q_0'}^{-1})(Q_0'D_0{Q_0'}^{-1}),$$

this is the desired Polar Decomposition because $Q_0{Q_0'}^{-1}$ is orthogonal (as Q_0 and Q_0' are orthogonal, so ${Q_0'}^{-1}$ is orthogonal, and a product of orthogonal matrices is always orthogonal) and $Q_0'D_0{Q_0'}^{-1} = Q_0'D_0{Q_0'}^\top$ is positive-definite and symmetric due to the same properties for the diagonal D_0: this deduction goes exactly like the calculation we did for S' near the start of the proof (e.g., $q_{Q_0'D_0{Q_0'}^\top}(\mathbf{x}) = q_{D_0}({Q_0'}^\top\mathbf{x}) > 0$ for all nonzero $\mathbf{x} \in \mathbf{R}^n$). This completes the construction of Q and S, by defining $Q = Q_0{Q_0'}^{-1}$ and $S = Q_0'D_0{Q_0'}^{-1}$.

Step 2. It remains to show that if also $A = Q_1S_1$ for orthogonal Q_1 and positive-definite symmetric S_1 then necessarily $Q_1 = Q$ and $S_1 = S$. It suffices to show $S_1 = S$, as then $QS = A = Q_1S_1 = Q_1S$, so multiplying both sides by S^{-1} on the right would yield that $Q = Q_1$. To show that $S_1 = S$, we will again exploit the Spectral Theorem. The key observation is that the contributions of Q and Q_1 to $A^\top A$ disappear: since $Q^\top = Q^{-1}$ by orthogonality of Q and $S^\top = S$ by symmetry of S, we have

$$A^\top A = (QS)^\top(QS) = S^\top Q^\top QS = SQ^{-1}QS = S^2$$

and similarly $A^\top A = S_1^2$ by using the alternative expression Q_1S_1 for A. Putting it all together, we have $S^2 = S_1^2$, and from this we shall deduce that $S = S_1$.

Rather generally, we shall show that if S and T are positive-definite symmetric $n \times n$ matrices for which $S^m = T^m$ for some positive integer m then $S = T$ (we then apply it with $T = S_1$ and $m = 2$). This property of positive-definite symmetric matrices (which is false for general invertible $n \times n$ matrices) is an analogue of the familiar fact for positive scalars s and t that if $s^m = t^m$ then $s = t$ (and more specifically: if $0 < s < t$ then $s^m < t^m$). We will use the Spectral Theorem to deduce this matrix assertion from the scalar analogue (applied to eigenvalues)! The general principle of which this is a special case is that the Spectral Theorem enables many constructions and arguments with positive real numbers to be generalized to the setting of positive-definite symmetric matrices.

---

[28] Elie Cartan (1869-1951) was was one of the most important mathematicians in the first half of the 20th century, making fundamental discoveries in partial differential equations, geometry, and the study and classification of "continuous symmetry". A school inspector visiting his elementary school noticed his exceptional talent and set him on the path to scholarships and a university education, which was usually out of reach at that time in France for children from very poor families such as his.

The first key point is to check that S^m is positive-definite and symmetric (and the same then applies to T^m by using T instead of S). It is generally not true that matrix multiplication preserves symmetry or positive-definiteness, but mth powers are rather special among matrix products. For example, S^m is symmetric because the "flip" interaction of matrix multiplication and transpose yields $(S^m)^\top = (S^\top)^m = S^m$. Positive-definiteness lies a bit deeper: it relies on the Spectral Theorem for S. That result provides an orthogonal basis $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ of $\mathbf{R}^n$ consisting of eigenvectors for S, and $S\mathbf{v}_j = \lambda_j \mathbf{v}_j$ with $\lambda_j > 0$ by positive-definiteness of S (Proposition 24.2.10(i)). For later purposes we scale the $\mathbf{v}_j$'s so that they are all unit vectors, and we arrange the labels so that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. As in our considerations with matrix powers and eigenvectors in (24.3.1), we have $S^m \mathbf{v}_j = \lambda_j^m \mathbf{v}_j$ for all j. Hence, the orthogonal basis $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ of $\mathbf{R}^n$ also consists of eigenvectors for S^m, with eigenvalues λ_j^m that are all positive. Applying Proposition 24.2.10(i) to the symmetric S^m, we conclude from the positivity of its eigenvalues that S^m is positive-definite!

Step 3. Next, we harness some ideas from the proof of the Spectral Theorem in Section B.3 (applied to both S and S^m, as well as to both T and T^m). In that proof we built an eigenvector by maximizing a specific quadratic form on the unit sphere in $\mathbf{R}^n$. More specifically, we showed for any symmetric $n \times n$ matrix M that any unit vector at which $q_M(\mathbf{x}) = \mathbf{x} \cdot (M\mathbf{x})$ attains a maximum on the unit sphere *must* be an eigenvector for M, with eigenvalue equal to the maximal value.

Where is $q_{S^m}(\mathbf{x})$ maximized on the unit sphere in $\mathbf{R}^n$? We claim that this occurs exactly where $q_S(\mathbf{x})$ is maximized! Although $q_{S^m}(\mathbf{x})$ and $q_S(\mathbf{x})^m$ typically have nothing to do with each other, at certain unit vectors these will match in a useful way. To see what is going on, we compute everything in terms of the orthonormal basis $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ of eigenvectors for S. Since this is an orthonormal basis (not just an orthogonal basis) and λ_n is the largest eigenvalue (and all eigenvalues are positive), the formula (24.2.2) applied to S^m says that for any $\mathbf{v} = \sum_{j=1}^n t_j \mathbf{v}_j \in \mathbf{R}^n$ we have

$$q_{S^m}(\mathbf{v}) = \sum_{j=1}^n \lambda_j^m t_j^2 \leq \sum_{j=1}^n \lambda_n^m t_j^2 = \lambda_n^m \sum_{j=1}^n t_j^2 = \lambda_n^m \|\mathbf{v}\|^2 \tag{B.4.2}$$

(the final equality using that the $\mathbf{v}_j$'s are an orthonormal basis, which gives that $\mathbf{v} \cdot \mathbf{v} = \sum_{j=1}^n t_j^2$; keep in mind that the t_j's are not defined as the entries of the n-vector $\mathbf{v}$ but rather as the coefficients when $\mathbf{v}$ is expressed in terms of the orthonormal basis of $\mathbf{v}_j$'s). Letting $\lambda = \lambda_n$ be the largest eigenvalue of S, in the special case that $\mathbf{v}$ is a unit vector we conclude from (B.4.2) that $q_{S^m}(\mathbf{v}) \leq \lambda^m$ *and* that the upper bound λ^m is attained at *exactly* those unit vectors $\mathbf{v}$ for which the inequalities $\lambda_j^m t_j^2 \leq \lambda^m t_j^2$ are *equalities* for all j. For j with $\lambda_j < \lambda$ we have $\lambda_j^m < \lambda^m$ (as λ_j and λ are both positive), so such equality occurs only when $t_j = 0$. On the other hand, if $\lambda_j = \lambda$ then such equality holds for any t_j. Hence, q_{S^m} is maximized on the unit sphere at *exactly* those unit vectors $\mathbf{v}$ in the span of the $\mathbf{v}_j$'s having the maximal S-eigenvalue λ. We shall now describe this span in a more useful way.

The eigenvalues have been arranged in monotonically increasing order, so $\lambda_j = \lambda$ precisely for $j = r, r+1, \dots, n$ for some $r \leq n$. Hence, an expression $\mathbf{v} = \sum_{j=r}^n t_j \mathbf{v}_j$ as a span of such $\mathbf{v}_j$'s implies

$$S\mathbf{v} = \sum_{j=r}^n t_j (S\mathbf{v}_j) = \sum_{j=r}^n t_j (\lambda \mathbf{v}_j) = \lambda \sum_{j=r}^n t_j \mathbf{v}_j = \lambda \mathbf{v},$$

so $\mathbf{v}$ is then an S-eigenvector with eigenvalue λ. The same works in reverse: if $\mathbf{v}$ is a unit vector (or even any vector) satisfying $S\mathbf{v} = \lambda \mathbf{v}$ then necessarily $\mathbf{v}$ is in the span of the $\mathbf{v}_j$'s for which $\lambda_j = \lambda$. Indeed, writing $\mathbf{v} = \sum_{j=1}^{n} t_j\mathbf{v}_j$ for some scalars $t_1, \dots, t_n$, we have

$$S\mathbf{v} = \sum_{j=1}^{n} t_j(S\mathbf{v}_j) = \sum_{j=1}^{n} t_j(\lambda_j\mathbf{v}_j) = \sum_{j=1}^{n}(\lambda_j t_j)\mathbf{v}_j, \;\; \lambda\mathbf{v} = \lambda\sum_{j=1}^{n} t_j\mathbf{v}_j = \sum_{j=1}^{n}(\lambda t_j)\mathbf{v}_j,$$

and the equality of $S\mathbf{v}$ and $\lambda\mathbf{v}$ implies the equality termwise for these sums since the $\mathbf{v}_j$'s are *linearly independent*. Equating corresponding coefficients then gives $\lambda_j t_j = \lambda t_j$ for all j, so when $\lambda_j < \lambda$ (i.e., $j < r$) necessarily $t_j = 0$; this says $\mathbf{v} = \sum_{j=r}^{n} t_j\mathbf{v}_j$, as desired.

To summarize: $q_{S^m}(\mathbf{x})$ is maximized on the unit sphere in $\mathbf{R}^n$ at *exactly* the unit vectors that are S-eigenvectors with the maximal S-eigenvalue. But $S^m = T^m$ by assumption, and the preceding reasoning with S applies equally well to T, so the quadratic form $q_{S^m}(\mathbf{x}) = q_{T^m}(\mathbf{x})$ is also maximized on the unit sphere in $\mathbf{R}^n$ at *exactly* the unit vectors that are T-eigenvectors with maximal T-eigenvalue. Voila: this shows that the maximal S-eigenvalue and maximal T-eigenvalue coincide and that the unit eigenvectors for that common eigenvalue are the *same*. Every eigenvector is a scalar multiple of a unit eigenvector (since eigenvectors are nonzero), so we conclude that S and T share the *same* maximal eigenvalue λ and that the λ-eigenvectors for S are the *same* as those for T. Observe how the Spectral Theorem led us to such conclusions about S and T from the assumption of equality of S^m and T^m.

Step 4. Since $S\mathbf{v}_n = \lambda\mathbf{v}_n$, we conclude that $\mathbf{v}_n$ is an eigenvector for both S and T with the same eigenvalue (namely $\lambda = \lambda_n$) for each. We aim to show the same for all $\mathbf{v}_j$'s, so then we would have $S = T$ since any $\mathbf{v} = \sum_{j=1}^{n} t_j\mathbf{v}_j \in \mathbf{R}^n$ would then satisfy

$$S\mathbf{v} = \sum_{j=1}^{n} t_j(S\mathbf{v}_j) = \sum_{j=1}^{n} t_j(\lambda_j\mathbf{v}_j) = \sum_{j=1}^{n} t_j(T\mathbf{v}_j) = T\mathbf{v}$$

(forcing $S = T$ as desired). To bootstrap from the largest eigenvalue to the rest, we will pass to lower-dimensional subspaces exactly as in the proof of the Spectral Theorem.

The subspace $H = \operatorname{span}(\mathbf{v}_1, \dots, \mathbf{v}_{n-1})$ is the orthogonal complement to $\mathbf{v}_n$ due to the $\mathbf{v}_j$'s being an orthogonal basis, and since $\mathbf{v}_n$ is an eigenvector for both S and T it follows that H is preserved by both S and T due to the symmetry of each: if $\mathbf{x} \cdot \mathbf{v}_n = 0$ then

$$(T\mathbf{x}) \cdot \mathbf{v}_n = \mathbf{x} \cdot (T\mathbf{v}_n) = \mathbf{x} \cdot (\lambda\mathbf{v}_n) = \lambda(\mathbf{x} \cdot \mathbf{v}_n) = \lambda(0) = 0$$

and similarly for S. In particular, it makes sense to consider the effects of S and T on H as functions $H \to H$.

Exactly as in Step 1 of the proof of the Spectral Theorem, if we identify H with $\mathbf{R}^{n-1}$ by using coefficients relative to an orthonormal basis of H then the effects of S and T on H become symmetric linear functions $\mathbf{R}^{n-1} \to \mathbf{R}^{n-1}$. Moreover, these $(n-1) \times (n-1)$ symmetric matrices are positive-definite because their associated quadratic forms are exactly the effects of q_S and q_T on H (which are positive away from the origin due to the positive-definiteness of S and T on the entirety of $\mathbf{R}^n$). Also, the mth powers of these $(n-1) \times (n-1)$ matrices are equal because they are each the effect of $S^m = T^m$ on H.

Putting it all together, we conclude that the study of the effects of S and T on H is an $(n-1)$-dimensional version of our same overall problem. Thus, by induction on dimension we conclude that S and T coincide on H. But every $\mathbf{v} \in \mathbf{R}^n$ can be written as $\mathbf{v} = \mathbf{h} + c\mathbf{v}_n$ for some $\mathbf{h} \in H$ and scalar c (express $\mathbf{v}$ as a linear combination of the $\mathbf{v}_j$'s and collect the contribution from $\mathbf{v}_1, \dots, \mathbf{v}_{n-1}$ to get $\mathbf{h}$), so the equality of S and T on both the linear subspace H and on the vector $\mathbf{v}_n$ gives

$$S\mathbf{v} = S(\mathbf{h} + c\mathbf{v}_n) = S\mathbf{h} + c(S\mathbf{v}_n) = T\mathbf{h} + c(T\mathbf{v}_n) = T(\mathbf{h} + c\mathbf{v}_n) = T\mathbf{v}.$$

This shows that $S = T$, as desired, and so completes the proof of the Polar Decomposition. ☐

> "There is no royal road to geometry."
>
> Euclid (allegedly said to King Ptolemy I)

C. Equivalence of two perspectives on ellipses and hyperbolas (optional)

Ellipses and hyperbolas are often defined by equations of the form $Ax^2 + By^2 = C$ with $A, B, C \neq 0$. But the concepts of "ellipse" and "hyperbola" were originally defined without equations in ancient Greek geometry, by Menaechmus around 350 BC, via slicing a cone with a plane (so such curves are called "conic sections"). The ancient Greeks did not have the algebraic language that we take for granted, nor the concepts of negative number or 0, nor coordinate geometry (an idea that emerged in the 17th century). But they knew how to go from cone-based definitions to the equations of ellipses and hyperbolas via results expressed in an entirely different way (see [**Arch**, pp. 147-150], [**Apol**, Prop. 2, Prop. 3, pp. 9-12]).

The ancient Greek geometers also knew that the cone-based definitions of "ellipse" and "hyperbola" are equivalent to planar definitions in terms of distance conditions [**Apol**, Prop. 73, p. 118][29]. This appendix directly shows that the planar definitions in terms of distance conditions (provided below) are equivalent to definitions in terms of equations, bypassing the intervention of cones.

C.1. Ellipses. The geometric planar definition of an *ellipse* E is the collection of points in a plane whose distances to a chosen pair of points P, Q (called "foci") have sum equal to a fixed positive number larger than the distance PQ. See Figure C.1.1 for a picture, with green and orange segments illustrating the distance condition. Our aim is to express E via an equation of the form $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ for some $a, b > 0$.

Denote by $2a$ the fixed positive sum of distances to P and Q in the definition of E, so $2a > PQ$ (see Figure C.1.1). Set up coordinates so the midpoint of the segment $\overline{PQ}$ is (0,0), with P and Q on the x-axis; relabel if necessary so P is on the non-negative x-axis and Q is on the non-positive x-axis: $P = (c, 0)$ and $Q = (-c, 0)$ where $2c \geq 0$ is the length of $\overline{PQ}$. (The case $P = Q$ is allowed; ellipses with equal foci are circles by another name.) Hence $2a > 2c$, so $a > c$.

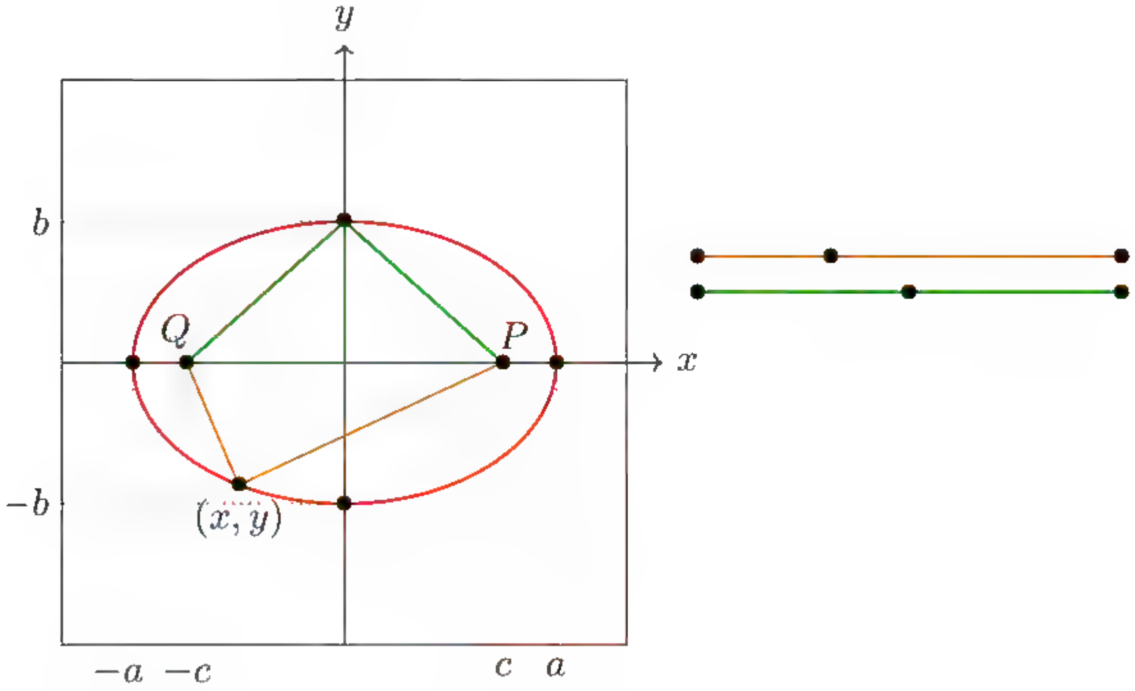


FIGURE C.1.1. The points on the red ellipse with foci at $(\pm c, 0)$ have their sum of distances to the foci (e.g., green and orange paths) equal to $2a$, with $a > c$.

---

[29] An especially elegant proof of this equivalence using inscribed spheres was found in 1822 by G. Dandelin [**Dan**].

Denote the y-intercepts of E as $(0, \pm b)$ with $b > 0$. Since $2a$ is the sum of the distances from each point on E to the two foci $(\pm c, 0)$ (by the way $2a$ was introduced at the start), the sum of the distances from each point $(0, \pm b)$ to $(c, 0)$ and $(-c, 0)$ is $2a$. Thus, by the Pythagorean Theorem $2a = \sqrt{b^2 + c^2} + \sqrt{b^2 + (-c)^2} = 2\sqrt{b^2 + c^2}$. Hence, $b^2 = a^2 - c^2$ (i.e., $b = \sqrt{a^2 - c^2} \leq a$).

The distance-defining condition for E is $\sqrt{(x-c)^2 + y^2} + \sqrt{(x+c)^2 + y^2} = 2a$. Writing this as

$$\sqrt{(x-c)^2 + y^2} = 2a - \sqrt{(x+c)^2 + y^2},$$

if we square both sides then we get

$$(x-c)^2 + y^2 = 4a^2 - 4a\sqrt{(x+c)^2 + y^2} + (x+c)^2 + y^2.$$

Expanding out the squares on the left and right, this says

$$x^2 - 2xc + c^2 + y^2 = 4a^2 - 4a\sqrt{(x+c)^2 + y^2} + x^2 + 2xc + c^2 + y^2.$$

Cancelling the common terms x^2, c^2, y^2 on both sides, this says

$$-2xc = 4a^2 - 4a\sqrt{(x+c)^2 + y^2} + 2xc,$$

or equivalently

$$4a\sqrt{(x+c)^2 + y^2} = 4a^2 + 4xc.$$

Dividing by $4a$ throughout, this says

$$\sqrt{(x+c)^2 + y^2} = a + x(c/a). \tag{C.1.1}$$

Now squaring both sides again, this says

$$(x+c)^2 + y^2 = a^2 + 2xc + x^2(c^2/a^2),$$

and expanding out the left side turns this into

$$x^2 + 2xc + c^2 + y^2 = a^2 + 2xc + x^2(c^2/a^2).$$

Cancelling $2xc$ from both sides and subtracting the x^2-term on the right over to the left and subtracting the constant term c^2 on the left over to the right (since $c^2 < a^2$) turns this into

$$x^2\left(1 - \frac{c^2}{a^2}\right) + y^2 = a^2 - c^2,$$

or equivalently

$$x^2\left(\frac{a^2 - c^2}{a^2}\right) + y^2 = a^2 - c^2.$$

But $a^2 - c^2 = b^2$, so dividing by b^2 throughout turns this into $x^2/a^2 + y^2/b^2 = 1$.

To go in reverse, showing that any (x, y) satisfying $x^2/a^2 + y^2/b^2 = 1$ also satisfies the distance-based definition of E (so the equation defines precisely the points of E), we just need to make sure that steps where quantities were squared can be undone without sign ambiguity in the formation of square roots. Looking back at the steps where both sides of an equation were squared, we have to verify the non-negativity of $a + x(c/a)$ and $2a - \sqrt{(x+c)^2 + y^2}$ for any (x, y) satisfying $x^2/a^2 + y^2/b^2 = 1$.

The equation implies $x^2/a^2 \leq 1$, so $|x| \leq a$. Hence, $a + x(c/a) \geq a - a(c/a) = a - c > 0$. To show $2a - \sqrt{(x+c)^2 + y^2} \geq 0$ is the same as to show $2a \geq \sqrt{(x+c)^2 + y^2}$. But the algebraic work above can be run in reverse starting from the equation $x^2/a^2 + y^2/b^2 = 1$ to deduce that (C.1.1) must hold (since we have just shown that $a + x(c/a) \geq 0$). Hence, it is equivalent to show that $2a \geq a + x(c/a)$. Subtracting a from both sides and then multiplying both sides by the positive a/c, it is the same as showing $a^2/c \geq x$. But $0 < c < a$, so $a^2/c > a^2/a = a \geq x$, as desired.

Remark C.1.1. The preceding work yields an equation for E with $b \leq a$ because we set up the coordinates with foci on the x-axis, and any $0 < b \leq a$ arises: use foci at $(\pm c, 0)$ for $c = \sqrt{a^2 - b^2}$. The equations with $a \leq b$ correspond to swapping the roles of x and y, so in effect setting up coordinates with foci on the y-axis. (Circles are the case with equal foci $P = Q$: $c = 0$ or equivalently $a = b$.)

C.2. Hyperbolas. The geometric planar definition of a *hyperbola* H is the collection of points in a plane whose distances to a chosen pair of distinct points P, Q (called "foci") have difference in the sense absolute value equal to a fixed positive number less than the distance PQ. See Figure C.2.1 for a picture, with green and orange segments illustrating the distance condition. Our next aim is to express H via an equation of the form $x^2/a^2 - y^2/b^2 = 1$ for some $a, b > 0$.

Denote by $2a$ the fixed positive absolute difference of distances to P and Q in the definition of H, so $2a < PQ$ (see Figure C.2.1). Set up coordinates in the plane so the midpoint of the segment $\overline{PQ}$ is (0,0), with P and Q on the x-axis. Relabel if necessary so P is on the positive x-axis and Q is on the negative x-axis: $P = (c, 0)$ and $Q = (-c, 0)$ where $2c > 0$ is the length of $\overline{PQ}$. Hence, $2a < 2c$, so $a < c$. In particular, it makes sense to *define* $b = \sqrt{c^2 - a^2} > 0$, so $b^2 = c^2 - a^2$.

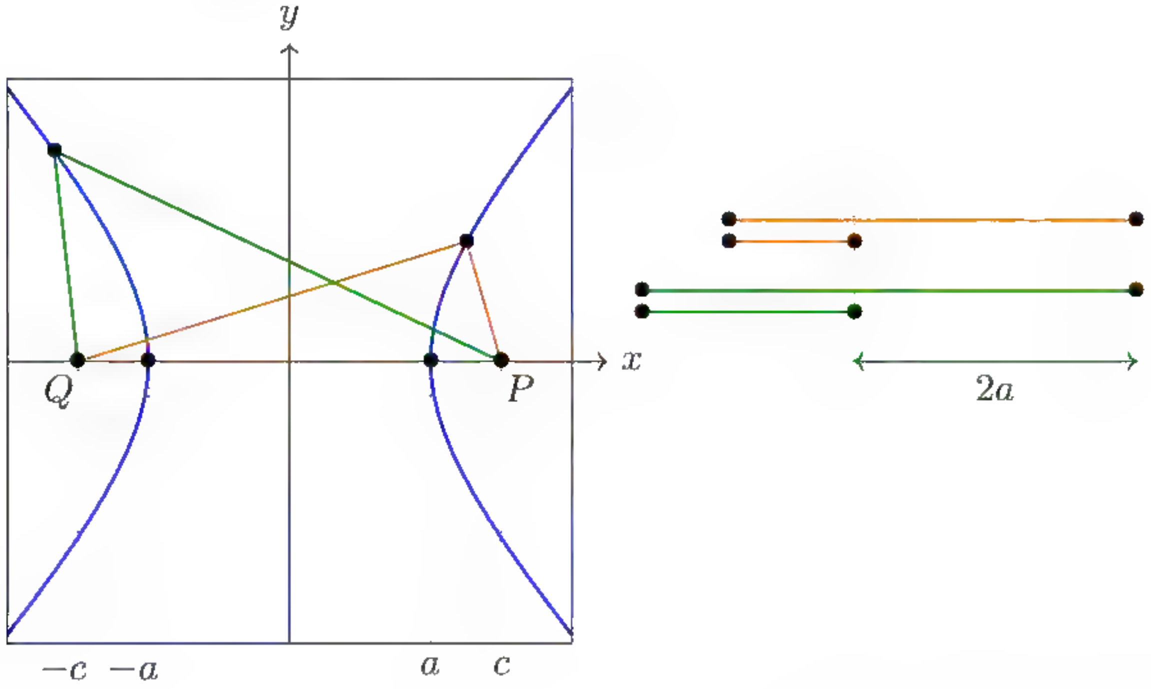


FIGURE C.2.1. The blue hyperbola H with foci at $(\pm c, 0)$ and difference of distances to them (in the sense of absolute value) equal to $2a$, with $a < c$.

Points on the y-axis are equidistant from P and Q due to how we set up coordinates, so the difference of their distances to P and Q is 0. But points on H have distances to P and Q whose difference has a fixed positive absolute value, so points $(x, y) \in H$ cannot lie on the y-axis and hence $x \neq 0$. If $x > 0$ (the "right half") then the distance-defining condition for H is

$$\sqrt{(x+c)^2 + y^2} - \sqrt{(x-c)^2 + y^2} = 2a;$$

it is the opposite order of subtraction when $x < 0$ (always bigger distance minus smaller distance). In other words, when $x < 0$ the effect in this latter equation is to swap the roles of c and $-c$.

Suppose for now that $x > 0$. Writing this as

$$\sqrt{(x+c)^2 + y^2} = 2a + \sqrt{(x-c)^2 + y^2},$$

squaring both sides yields

$$(x+c)^2 + y^2 = 4a^2 + 4a\sqrt{(x-c)^2 + y^2} + (x-c)^2 + y^2.$$

Expanding out the squares on the left and right, this says

$$x^2 + 2xc + c^2 + y^2 = 4a^2 + 4a\sqrt{(x-c)^2 + y^2} + x^2 - 2xc + c^2 + y^2.$$

Cancelling the common terms x^2, c^2, y^2 on both sides, this says

$$2xc = 4a^2 + 4a\sqrt{(x-c)^2 + y^2} - 2xc,$$

or equivalently

$$4a\sqrt{(x-c)^2 + y^2} = -4a^2 + 4xc.$$

Dividing by $4a$ throughout, this says

$$\sqrt{(x-c)^2 + y^2} = -a + x(c/a). \tag{C.2.1}$$

If instead $x < 0$, we obtain the analogous equation with c negated:

$$\sqrt{(x+c)^2 + y^2} = -a + x(-c/a).$$

Continuing with $x > 0$, squaring both sides of (C.2.1) yields

$$(x-c)^2 + y^2 = a^2 - 2xc + x^2(c^2/a^2),$$

and expanding out the left side turns this into

$$x^2 - 2xc + c^2 + y^2 = a^2 - 2xc + x^2(c^2/a^2).$$

Cancelling $-2xc$ from both sides and subtracting $x^2 + y^2$ on the left over to the right and subtracting the constant term a^2 on the right over to the left (since $a^2 > c^2$) turns this into

$$c^2 - a^2 = x^2\left(\frac{c^2}{a^2} - 1\right) - y^2,$$

and the same is obtained when $x < 0$ since in this final equality c and x only appears through their squares. Summarizing, in both cases (either sign for $x \neq 0$) we have

$$x^2\left(\frac{c^2 - a^2}{a^2}\right) - y^2 = c^2 - a^2.$$

But $c^2 - a^2 = b^2$, so dividing by b^2 throughout turns this into $x^2/a^2 - y^2/b^2 = 1$.

To go in reverse, showing that any (x, y) satisfying $x^2/a^2 - y^2/b^2 = 1$ also satisfies the distance-based definition of H (so the equation defines precisely the points of H), we just need to make sure that steps where quantities were squared can be undone without sign ambiguity in the formation of square roots, being attentive to the sign of x. The equation $x^2/a^2 - y^2/b^2 = 1$ implies $x^2/a^2 \geq 1$, so $|x| \geq a$. In particular, $x \neq 0$ (since $a > 0$). Looking back at the steps where both sides of an equation were squared in the case $x > 0$, we have to verify the non-negativity of $-a + x(c/a)$ and $2a + \sqrt{(x+c)^2 + y^2}$ for any (x, y) satisfying $x^2/a^2 - y^2/b^2 = 1$ with $x > 0$. If instead $x < 0$ then we have to verify the non-negativity of $-a + x(-c/a)$ and $2a + \sqrt{(x-c)^2 + y^2}$.

The sums $2a + \sqrt{(x \pm c)^2 + y^2}$ are non-negative since a sum of non-negative numbers is always non-negative. It remains to show $-a + x(c/a) \geq 0$ when $x > 0$ and $-a + x(-c/a) \geq 0$ when $x < 0$. Since $-x = |x|$ when $x < 0$, a unified assertion is that $-a + |x|(c/a) \geq 0$ always. We have seen that $|x| \geq a$, so $-a + |x|(c/a) \geq -a + a(c/a) = -a + c > 0$. This completes the argument.

Remark C.2.1. The preceding yields an equation for H in which x^2 has a positive coefficient $1/a^2$ and y^2 has a negative coefficient $-1/b^2$ because we set up the coordinates with foci on the x-axis, and any $a, b > 0$ arises: use foci at $(\pm c, 0)$ for $c = \sqrt{a^2 + b^2}$. Equations of the form $-x^2/a^2 + y^2/b^2 = 1$ (i.e., with coefficient signs swapped), or equivalently $x^2/a^2 - y^2/b^2 = -1$, correspond to swapping the roles of x and y, so in effect setting up the coordinates with foci on the y-axis.

"I make my money on the stock market. I don't make it by proving theorems."

C. Shannon

D. Google's PageRank algorithm (optional)

Go to a webpage and start clicking links at random. (One of the authors spends a regrettable amount of time doing exactly this.) Where do you end up? You will, of course, end up spending a lot more time on "important" websites than unimportant ones. This is related to the basic idea of Google's PageRank algorithm (named after Larry Page, not just "webpage"). We now explain a simplification of the original version of that algorithm.

D.1. The algorithm. When one does a Google search for a phrase, the output of the search algorithm is a list of (potentially hundreds of thousands of) webpages containing that phrase, and these webpages must be ordered in some way. The *method* of ordering the webpages containing the search phrase is the key innovation that made Google so successful in web search, and it involves assigning a "value" to each webpage (with the search outcome then arranged in decreasing order of this "value"). How is such a "value" to be determined?

Let n be the number of webpages on the Internet (a very large number, in the billions). For the ith webpage, Google seeks to assign it some value $v_i > 0$ with $\sum_{i=1}^{n} v_i = 1$ (i.e., the total value of all webpages is defined to be $100\% = 1$, with that total distributed across all webpages in some way to be determined). One of Google's first insights was that v_i should be determined by a method requiring no knowledge about the content of webpages, only about what links to what, trusting that the users set up links based on relevance and utility of content and that no webpage links to itself. The condition required on the v_i's is that each v_i should be a weighted sum of the values v_j for all j for which the jth webpage has a link to the ith one. In symbols, we want to find v_i's satisfying the (massive) system of n linear equations in n unknowns

$$v_i = \sum_{j=1}^{n} w_{ij} v_j \tag{D.1.1}$$

for certain specific numerical weights w_{ij} between 0 and 1 whose precise definition won't concern us here but which satisfy the following conditions:

(i) the definition of w_{ij} is given solely in terms of links of the ith and jth webpages to other webpages (so Google can calculate these coefficients once per month from a snapshot of the entire Internet in terms of links among webpages), and $w_{ij} = 0$ if the jth webpage doesn't link to the ith one.

(ii) the larger the value of w_{ij}, the more valuable is the link from the jth page to the ith page,

(iii) most w_{ij}'s are equal to 0 (since a given webpage tends not to be linked to a huge number of others relative to the overall size of the Internet) and the sum $\sum_{i=1}^{n} w_{ij}$ of the weights by which the jth page contributes to the value of *all* other pages to which it is linked is equal to 1 (a reasonable condition: the weights w_{ij} for a fixed j indicate *what fraction* of the value of the jth webpage contributes to the value of the ith one, so summing this over all i with j fixed should give $100\% = 1$).

In the language of vectors and matrices, if $\mathbf{v} \in \mathbf{R}^n$ is the vector whose ith entry is v_i (so we are trying to find $\mathbf{v}$) then (D.1.1) says

$$\mathbf{v} = W\mathbf{v}$$

where W is the $n \times n$ matrix (computed monthly by Google) whose ij-entry is w_{ij}. Keep in mind, as noted above, that we also want $v_i > 0$ for all i and $\sum_{i=1}^{n} v_i = 1$. Also, for any j, the sum of the entries in the jth column of W is

$$\sum_{i=1}^{n} w_{ij} = 1$$

(as noted in (iii) above). Thus, W is a Markov matrix.

How do we find $\mathbf{v}$? Although the equation $\mathbf{v} = W\mathbf{v}$, or equivalently $(\mathrm{I}_n - W)\mathbf{v} = \mathbf{0}$, can be regarded as a (massive) system of linear equations for which we seek a nonzero solution vector (having positive entries summing to 1), it is completely impractical – even for the most powerful computers – to solve this system directly. Indeed, with a billion webpages, if a computer can carry out a trillion multiplications every second then it would take around a billion years to solve $\mathbf{v} = W\mathbf{v}$ by direct means (such as via the classical "Gaussian elimination" method, essentially the LU-decomposition in Chapter 22).

Google made a small (but crucial) modification in the way they define the weights w_{ij}: they chose a specific number α slightly less than 1 (in fact, around 0.85) and replaced w_{ij} with

$$w'_{ij} = \alpha w_{ij} + \frac{1-\alpha}{n}$$

(recall that n is the number of webpages on the entire Internet, so $(1-\alpha)/n$ is extremely small). This adjustment prevents the PageRank algorithm from developing certain unrealistic features. Now, instead of many weights w_{ij} being equal to 0, they are all at least as large as the very tiny but positive number $(1-\alpha)/n$. Let W' be the $n \times n$ matrix whose ij-entry is w'_{ij}, so the column sums of W' also equal 1:

$$\sum_{i=1}^{n} w'_{ij} = \sum_{i=1}^{n} \left(\alpha w_{ij} + \frac{1-\alpha}{n} \right) = \alpha \sum_{i=1}^{n} w_{ij} + n \cdot \frac{1-\alpha}{n} = \alpha \cdot 1 + (1-\alpha) = 1.$$

Both W' and W are Markov matrices, as were the matrices denoted as M in Sections 16.1 and 16.2.

Sergey Brin and Larry Page [**BP**] found an excellent approximation to $\mathbf{v}$ by applying the following theorem on Markov matrices which had been proved in 1907 by Perron.[30]

Theorem D.1.1 (Perron). *If an $n \times n$ Markov matrix M has all entries positive then there is a* **unique** *vector $\mathbf{v} \in \mathbf{R}^n$ with positive entries summing to 1 for which $\mathbf{v} = M\mathbf{v}$.*

Moreover, if $\mathbf{v}_0$ is **any** *vector having non-negative entries summing to 1 then the vectors $M^k\mathbf{v}_0$ obtained by multiplying M repeatedly against $\mathbf{v}_0$ converge to $\mathbf{v}$ as $k \to \infty$. (Taking $\mathbf{v}_0$ to be any of the standard basis vectors $\mathbf{e}_i$, so $M^k\mathbf{e}_i$ is the ith column of M^k, it follows that for large k all columns of M^k are nearly equal to $\mathbf{v}$ and hence are all nearly the same as each other!)*

A proof of both the first part of Theorem D.1.1 and additional results from which the second part can be deduced with a bit more work is given in [**M**, Sec. 8.2]. (This reference has a more general setup – it doesn't assume M is Markov – and correspondingly a weaker-looking conclusion "$r\mathbf{v} = M\mathbf{v}$" for a special $r > 0$. But for Markov M it follows by summing up the entries in $r\mathbf{v}$ and $M\mathbf{v}$ that $r = 1$ in such cases, so Perron's theorem is obtained.)

In the setting of Theorem D.1.1 relevant to Google, we can take $\mathbf{v}_0 \in \mathbf{R}^n$ to be the vector whose first entry is 1 and whose other entries are all equal to 0 (or whose entries are all equal to $1/n$) and compute $W'^k\mathbf{v}_0$ for larger and larger values of k. Because of the way W' is made from the matrix W that has most of its entries equal to 0, despite the huge number of rows and columns the powers W'^k can be computed rather quickly (using specialized linear algebra techniques for working with "sparse matrices", such as W

---

[30] Oskar Perron (1880-1975) was a German mathematician who worked across a wide array of fields, including differential equations, linear algebra, geometry, and number theory, publishing papers well past the age of 80. He had a lifelong interest in the subject of continued fractions, on which he wrote a 2-volume masterpiece [**Pe**]. Perron's discovery of the tremendously useful Theorem D.1.1 grew out of his work on continued fractions [**Ha3**, Sec. 17.1].

that is very close to W') and so Google can solve for $\mathbf{v}$ in a matter of days via Perron's theorem: compute ${W'}^k\mathbf{v}_0$ for increasing k until it stabilizes to within very good accuracy.

D.2. A refinement of Perron's theorem. Although the positivity condition in Perron's Theorem D.1.1 is satisfied in the case relevant to Google, due to the switch from W to W', there are plenty of situations where a Markov matrix has lots of entries equal to 0. Both the bird migration example in Section 16.1 and the gambler's ruin example in Section 16.2 involved such a Markov matrix. When the positivity condition on all matrix entries in Perron's theorem is removed, the conclusion of the theorem can fail (see Remark 16.1.3 for such an example).

Frobenius[31] proved in 1912 that the situation with some entries equal to 0 is salvaged when the Markov matrix satisfies a property called "irreducibility" (which always holds in the positive setting of Perron's theorem); this generalization is called the *Perron–Frobenius theorem*. The Markov matrix in Section 16.1 is irreducible (and satisfies an even stronger condition called "primitive"), explaining why the columns of M^{20} in Section 16.1 – and even M^k for *all* bigger k – are the same to three decimal digits' accuracy. The 6×6 Markov matrix M in Section 16.2 is *not* irreducible, but Frobenius' work shows that M^k converges quickly as k grows (to a limit whose columns are not all the same!), so the vector of probabilities $\mathbf{p}(k) = M^k\mathbf{p}(0)$ in Section 16.2 is the same as $\mathbf{p}(100)$ in (16.2.2) to many decimal digits' accuracy no matter what big k we consider (even for $k \geq 30$).

In the realistic study of population dynamics for which one keeps track of births and deaths (which the bird migration example in Section 16.1 disregarded), a popular class of models involves powers of a type of non-Markov matrix called a *Leslie matrix* that has non-negative entries of which many are 0. The Perron–Frobenius theorem is applicable to Leslie matrices, leading to good qualitative predictions for long-term behavior in a variety of ecological or biological population problems.

Remark D.2.1. Theorem D.1.1 (along with Frobenius' generalization mentioned above) is an important result in the theory of dynamical systems, and Sergey Brin's father is an expert in dynamical systems; see [**BS**, 3.3, 4.12]. Although you can use Perron's Theorem D.1.1 without any knowledge of why it is true, if you want to understand why the theorem works then you'll have to learn more math. The systematic study of large powers of matrices (and a huge amount more in linear algebra) relies crucially on the concept of *eigenvalue*, for which we give an introduction in Chapters 23 and 24 and whose utility rests on the non-commutativity of matrix multiplication.

Most of our discussion of eigenvalues focuses on $n \times n$ matrices A symmetric around their diagonal ($a_{ij} = a_{ji}$) since that suffices for our primary needs in this book, but the matrix M in Theorem D.1.1 is typically not symmetric (e.g., in the application to PageRank the matrix is never symmetric, since the weights w_{ij} used in (D.1.1) are highly non-symmetric under swapping i and j: the value of a link from the jth page to the ith page has nothing at all to do with the value of a link from the ith page to the jth page). Moreover, even though the statement of Theorem D.1.1 involves matrices and vectors whose entries are real numbers, its proof involves linear algebra with *complex numbers*. So if you want to understand why the theorem is true then you'll need to learn more about eigenvalues than is covered in this book; Math 104 and Math 113 cover that further material.

---

[31] Georg Frobenius (1849-1917) was a German mathematician who made discoveries across the whole spectrum of pure mathematics as it existed in the second half of the 19th century, with his greatest work being in algebraic directions (see [**Ha3**]). He made important contributions to matrix algebra early in his career, and in his final period of research established "Perron–Frobenius theory" that gave a definitive analogue of Theorem D.1.1 for matrices with non-negative entries.

> "One cannot escape the feeling that these mathematical formulas have an independent existence and intelligence of their own, that they are wiser than we are, wiser than even their discoverers, that we get more out of them than was originally put into them."
>
> H. Hertz

E. General determinants (optional)

In this appendix, we describe a very useful function called the *determinant*. For most functions considered in this book, the input is a vector and the output is either a vector or a scalar. The determinant is a different kind of function: the input is an $n \times n$ matrix A and the output is a number called the *determinant* of A and written $\det(A)$.

There is an explicit formula defining $\det(A)$. However, unless n is very small, that formula has a *lot* of terms and is not particularly useful for calculating the determinant. So we will not give that formula here. Instead, we describe the geometric meaning of the determinant, the rules that the determinant obeys, and how to use those rules to calculate and work with the determinant.

As mentioned above, the input of the determinant is an $n \times n$ matrix A. But we can also view the input as n vectors in $\mathbf{R}^n$, namely the columns of the matrix A. Thus if $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ are the columns of A, then $\det(A)$ and $\det(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n)$ are two notations for the same quantity. Another common notation for the determinant A is to replace brackets with vertical lines, so the following all denote the same thing:

$$\det \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} = \det \left(\begin{bmatrix} 1 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 4 \end{bmatrix} \right).$$

E.1. Properties of the determinant. We want to state the essential properties that the determinant satisfies, and then illustrate through a series of examples how to use these properties to compute determinants. The procedures arising in these properties are generally called *row and column operations* since they involve modifying rows or columns of an $n \times n$ matrix in specific ways.

Before stating these properties, we need to introduce a bit of terminology. The *diagonal* of an $n \times n$ matrix A consists of the entries a_{ii} going from the upper left corner (i.e., a_{11}) to the lower right corner (i.e., a_{nn}). If all entries **below** the diagonal are 0 then the matrix is called *upper triangular*, and if all entries **above** the diagonal are 0 then the matrix is called *lower triangular*. (These notions are encountered in Example 15.1.7 and Example 22.1.1.) We call A *triangular* if it is upper triangular or lower triangular.

For example, in the 3×3 case the following types of matrices

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix}, \quad \begin{bmatrix} b_{11} & 0 & 0 \\ b_{21} & b_{22} & 0 \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

are respectively upper triangular and lower triangular. In contrast, here is a **non-triangular** matrix:

$$\begin{bmatrix} 0 & 0 & 2 \\ 0 & 3 & 4 \\ 5 & 6 & 7 \end{bmatrix}.$$

Here are the main properties of determinants with respect to "row and column operations":

(1) The determinant of a triangular matrix is the product of the diagonal entries.
(2) Adding a multiple of one row of A to another row doesn't change $\det(A)$; likewise for columns.
(3) Switching two rows (or two columns) of A multiplies the determinant by -1.
(4) Multiplying a row (or a column) of A by a scalar multiplies the determinant by that scalar.

(5) If we fix all but one column of A, then $\det(A)$ is a linear function of the remaining column. For instance, in the case of column 1 for a 3×3 matrix we have

$$\det(\mathbf{w}_1 + \mathbf{w}_1', \mathbf{w}_2, \mathbf{w}_3) = \det(\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3) + \det(\mathbf{w}_1', \mathbf{w}_2, \mathbf{w}_3),$$
$$\det(c\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) = c\det(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3).$$

(6) Similarly, if we fix all but one row of A, then $\det(A)$ is a linear function of the remaining row.

Example E.1.1. Property (1) in the case of 3×3 matrices says

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{vmatrix} = a_{11}a_{22}a_{33}, \quad \begin{vmatrix} b_{11} & 0 & 0 \\ b_{21} & b_{22} & 0 \\ b_{31} & b_{32} & b_{33} \end{vmatrix} = b_{11}b_{22}b_{33}.$$

Also, although we have noted already that

$$\begin{bmatrix} 0 & 0 & 2 \\ 0 & 3 & 4 \\ 5 & 6 & 7 \end{bmatrix}$$

is not triangular, switching its first and third rows makes it triangular, so we can thereby compute its determinant using properties (3) and (1):

$$\begin{vmatrix} 0 & 0 & 2 \\ 0 & 3 & 4 \\ 5 & 6 & 7 \end{vmatrix} = (-1)\begin{vmatrix} 5 & 6 & 7 \\ 0 & 3 & 4 \\ 0 & 0 & 2 \end{vmatrix} = (-1)5 \cdot 3 \cdot 2 = -30.$$

■

In general, by using the properties (1)-(6) and some experience (illustrated in examples below) one can apply row and column operations (e.g., add or subtract multiples of one row or column from others to cancel out many entries to get lots of 0's) to quickly transform an $n \times n$ matrix to become triangular and thereby calculate its determinant. This requires accuracy with the arithmetic, but a computer can handle that extremely well (and a human can do it for $n \leq 3$, but even for $n = 3$ a computer is often more reliable than a human).

Example E.1.2. Let's calculate the determinant of $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 5 \\ 3 & 6 & 2 \end{bmatrix}$. By subtracting row 1 from row 2, and then 3 times row 1 from row 3, we do not change the determinant but we get some helpful 0's and can continue further to reach the triangular state:

$$\begin{aligned} \begin{vmatrix} 1 & 1 & 1 \\ 1 & 3 & 5 \\ 3 & 6 & 2 \end{vmatrix} &= \begin{vmatrix} 1 & 1 & 1 \\ 0 & 2 & 4 \\ 0 & 3 & -1. \end{vmatrix} \\ &= 2\begin{vmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 3 & -1. \end{vmatrix} && \text{(factoring a 2 from row 2)} \\ &= 2\begin{vmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & -7 \end{vmatrix} && \text{(3(row 2) subtracted from row 3)} \\ &= -14. \end{aligned}$$

■

Example E.1.3. Let's calculate the determinant of $\begin{bmatrix} 0 & 2 & 6 \\ 1 & 4 & 5 \\ 1 & 8 & a \end{bmatrix}$ for a general scalar a in the lower right entry. Using row and column operations, we have:

$$\begin{aligned}
\begin{vmatrix} 0 & 2 & 6 \\ 1 & 4 & 5 \\ 1 & 8 & a \end{vmatrix} &= (-1) \begin{vmatrix} 1 & 4 & 5 \\ 0 & 2 & 6 \\ 1 & 8 & a \end{vmatrix} && \text{(rows 1 and 2 switched)} \\
&= (-1) \begin{vmatrix} 1 & 4 & 5 \\ 0 & 2 & 6 \\ 0 & 4 & a-5 \end{vmatrix} && \text{(row 1 subtracted from row 3)} \\
&= (-1) \begin{vmatrix} 1 & 4 & 5 \\ 0 & 2 & 6 \\ 0 & 0 & a-17 \end{vmatrix} && \text{(2(row 2) subtracted from row 3)} \\
&= (-1)(1)(2)(a-17) \\
&= 34 - 2a.
\end{aligned}$$

■

In the preceding examples, there were lots of ways we could have proceeded: the choice of which row and column operations to use to arrive at a triangular matrix permits much flexibility. Consequently, it might be surprising that one really always arrives at the same final answer no matter how one chooses among the row and column operations to apply (assuming one hasn't made an arithmetic mistake). In other words, it may be surprising that the properties (1)–(6) of determinants under row and column are internally consistent. If you take a more advanced course in linear algebra then you will learn the general definition of determinants (which we have omitted in favor of the rules (1)–(6) listed above), from which one can establish all of the properties, or see [Ap, Ch. 3] (especially [Ap, Sec. 3.13]).

Theorem E.1.4. If an $n \times n$ matrix A has an entire row of 0's or an entire column of 0's then $\det(A) = 0$.

To see why this is true, note that if some column $\mathbf{v}$ is the zero vector then $\mathbf{v} = 0\,\mathbf{v}$. Thus, by property (5) we can factor out a 0 from that column to get $\det(A) = 0 \cdot \det(A) = 0$. The same argument applies using property (6) when there is a row that is the zero vector.

Remark E.1.5. The method of changing rows by adding (or subtracting) multiples of another row – which has no effect on $\det(A)$ – and then swapping rows if necessary (which can change $\det(A)$ by a sign) can always be used to bring ourselves to a triangular (and even diagonal) matrix. In this way, row operations *always* allow us to compute $\det(A)$ (and we can do similarly with columns if we prefer).

The main point is that if some column

$$\mathbf{v} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

has a nonzero entry a_i in the ith position then for any $j \neq i$ if we subtract "a_j/a_i times the ith row" from the jth row then we turn the jth entry in that column into 0. Doing this for all $j \neq i$ turns this column entirely into 0's except for the entry a_i. We refer to this as using the entry a_i in $\mathbf{v}$ as a "pivot" to make the rest of this column vanish.

For example, if we are given the matrix

$$A = \begin{bmatrix} 2 & -1 & 7 \\ 5 & 3 & 1 \\ -3 & 2 & 4 \end{bmatrix}$$

and focus on the first column, here is how to use its nonzero entry 5 in the second row to make all other entries in the first column vanish: we subtract $2/5$ times the second row from the first row to make the upper left entry vanish, and we subtract $-3/5$ times the second row from the third row to make the lower left entry vanish:

$$\begin{aligned}
\begin{vmatrix} 2 & -1 & 7 \\ 5 & 3 & 1 \\ -3 & 2 & 4 \end{vmatrix} &= \begin{vmatrix} 0 & -1-3(2/5) & 7-1(2/5) \\ 5 & 3 & 1 \\ -3 & 2 & 4 \end{vmatrix} && \text{(subtract $(2/5)$(row 2) from row 1)} \\
&= \begin{vmatrix} 0 & -11/5 & 33/5 \\ 5 & 3 & 1 \\ -3 & 2 & 4 \end{vmatrix} && \text{(carry out arithmetic in row 1)} \\
&= \begin{vmatrix} 0 & -11/5 & 33/5 \\ 5 & 3 & 1 \\ 0 & 2-(-3/5)3 & 4-(-3/5)1 \end{vmatrix} && \text{(subtract $(-3/5)$(row 2) from row 3)} \\
&= \begin{vmatrix} 0 & -11/5 & 33/5 \\ 5 & 3 & 1 \\ 0 & 19/5 & 23/5 \end{vmatrix} && \text{(carry out arithmetic in row 3)}
\end{aligned}$$

We can do this procedure with every column, so we either arrive at a step in which there is a column of 0's (so $\det(A) = 0$) or we clear out all entries except one in every column (as in the preceding calculation for the first column). The unique nonzero entry in each column occurs in some position.

Suppose as we vary across all columns the row in which each nonzero entry occurs never repeats (e.g., if the first column has its nonzero entry in the third row, then in no other column is the nonzero entry in the third row). By rearranging the rows (at the cost of possibly changing the determinant by a sign), we can therefore arrange that these nonzero entries occur along the diagonal, so we have a triangular (and even diagonal) matrix.

Suppose instead that some row contains the unique nonzero entry in *at least two* columns. Then we claim that there must be *some* row consisting entirely of 0's (so $\det(A) = 0$). Indeed, if some row is occupied by at least two of the nonzero entries then there are only $n-2$ other nonzero entries remaining yet there are $n-1$ other rows, so not all of those other rows can have a nonzero entry (there are too many other rows).

E.2. Formula for 2×2 determinants. As a further application of the properties of determinants, in the case of 2×2 matrices we can compute them in general, deducing that the notion of "determinant" introduced in the 2×2 case for the study of eigenvalues and eigenvectors in Theorem 23.3.1 coincides with the case $n = 2$ of $n \times n$ determinants as discussed in this appendix.

Theorem E.2.1. The determinant of $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is $ad - bc$.

To establish this formula, we first observe that as "row vectors"

$$\begin{bmatrix} a & b \end{bmatrix} = \begin{bmatrix} a & 0 \end{bmatrix} + \begin{bmatrix} 0 & b \end{bmatrix},$$

so by linearity in the first row (property (6)) we get

$$\begin{aligned}\begin{vmatrix} a & b \\ c & d \end{vmatrix} &= \begin{vmatrix} a & 0 \\ c & d \end{vmatrix} + \begin{vmatrix} 0 & b \\ c & d \end{vmatrix} \\ &= ad + \begin{vmatrix} 0 & b \\ c & d \end{vmatrix} \\ &= ad - \begin{vmatrix} b & 0 \\ d & c \end{vmatrix} && \text{(columns 1 and 2 switched)} \\ &= ad - bc.\end{aligned}$$

E.3. Relation of determinant to area and volume. Up to now, determinants may look like a peculiar algebraic concept. But in fact they have a rather compelling geometric meaning that underlies their importance in some applications. To explain this, we need to use the fact from Section 14.3 that any $n \times n$ matrix A can be thought as a linear function $T_A : \mathbf{R}^n \to \mathbf{R}^n$: given an input $\mathbf{x}$, the output is $A\mathbf{x}$. (See Definition 14.3.1.)

Example E.3.1. Let $B = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$. If the input is $\begin{bmatrix} x \\ y \end{bmatrix}$, the output is $B \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2x \\ y \end{bmatrix}$, so the associated linear function $T_B : \mathbf{R}^2 \to \mathbf{R}^2$ doubles the x-coordinate and leaves the y-coordinate unchanged.

Hence, if we apply the function T_B to any region U in $\mathbf{R}^2$, this region is stretched out horizontally by a factor of 2. Thus the area of the stretched region $T_B(U)$ (which we also denote as $B(U)$) should be twice the area of U (see Figure E.3.1). We therefore say that *the area magnification factor* of B is 2.

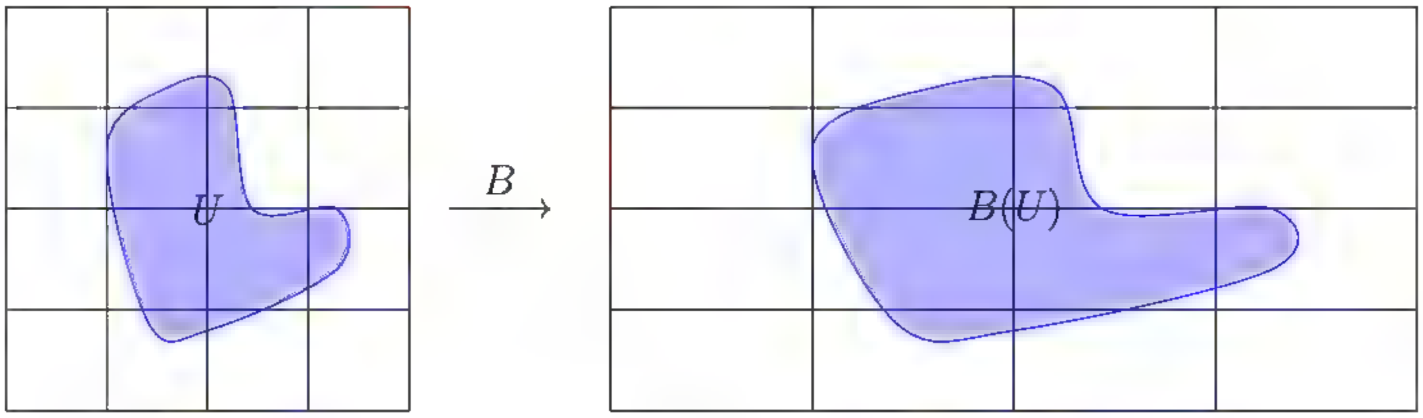


FIGURE E.3.1. The effect of B on a planar region U.

Consider $C = \begin{bmatrix} -2 & 0 \\ 0 & 1 \end{bmatrix}$. If the input is $\begin{bmatrix} x \\ y \end{bmatrix}$, the output is $\begin{bmatrix} -2x \\ y \end{bmatrix}$. If we apply the function T_C to a region U in $\mathbf{R}^2$, the region gets stretched out horizontally by a factor of 2 and then *flipped* around the y-axis (due to the minus sign). Flipping doesn't affect area, so the area of the stretched out region $C(U)$ is twice the area of U (see Figure E.3.2). Thus, the area magnification factor of C is 2, as for B.

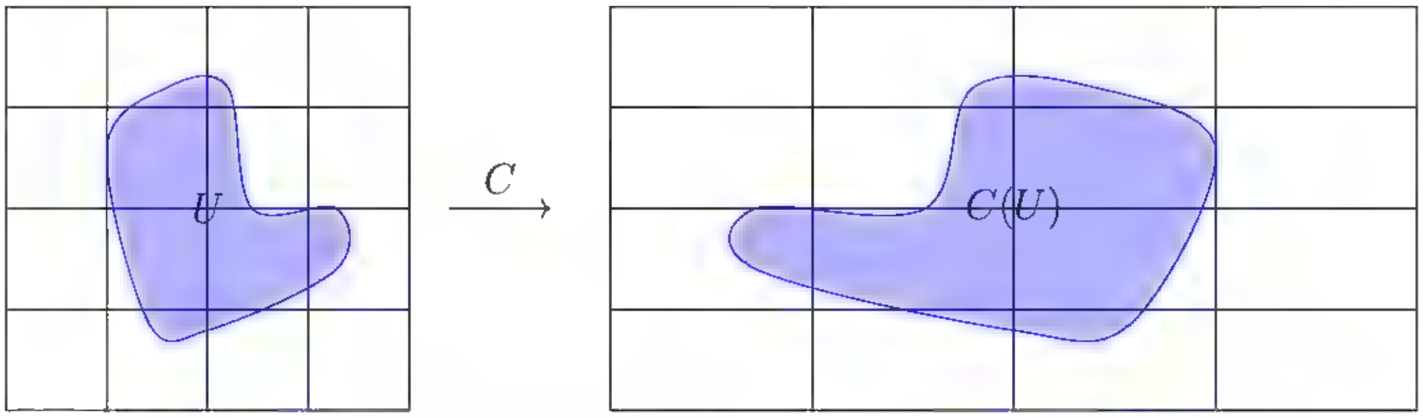


FIGURE E.3.2. The action of C on a planar region U.

Though B and C have the same area magnification factor (namely 2), there is something different about B and C. Figure E.3.3 shows a portrait of one of the members of the Stanford Math Department holding a koala bear.

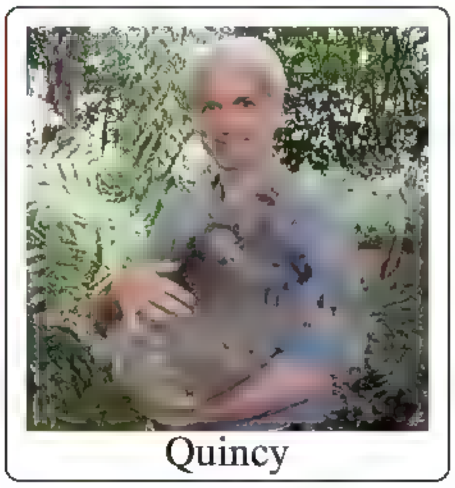


FIGURE E.3.3. Quincy

To maintain anonymity, we will call this person by the nickname Quincy. Note that Quincy has his *right* hand on top of the koala bear. Figures E.3.4 and E.3.5 show the result of feeding the picture of Quincy into the linear transformations T_B and T_C. After applying transformation T_B we still see Quincy's right hand on the koala, but after we apply transformation T_C we then see what appears to be Quincy's left hand on top of the koala (from the perspective of someone who only sees the output)!

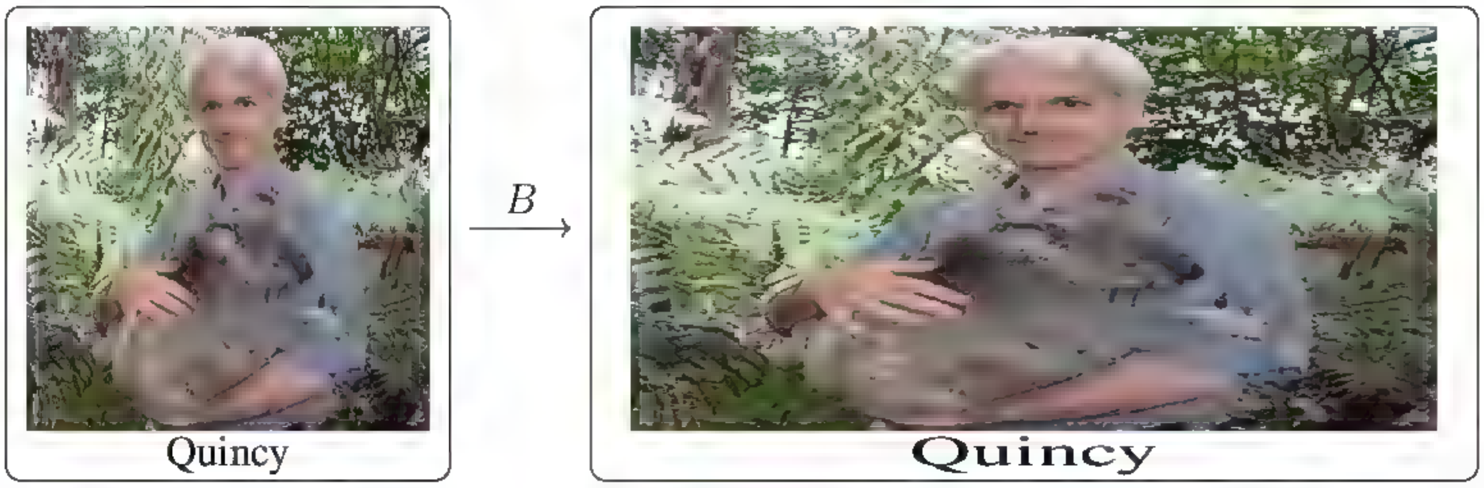


FIGURE E.3.4. The effect of B on the portrait of Quincy.

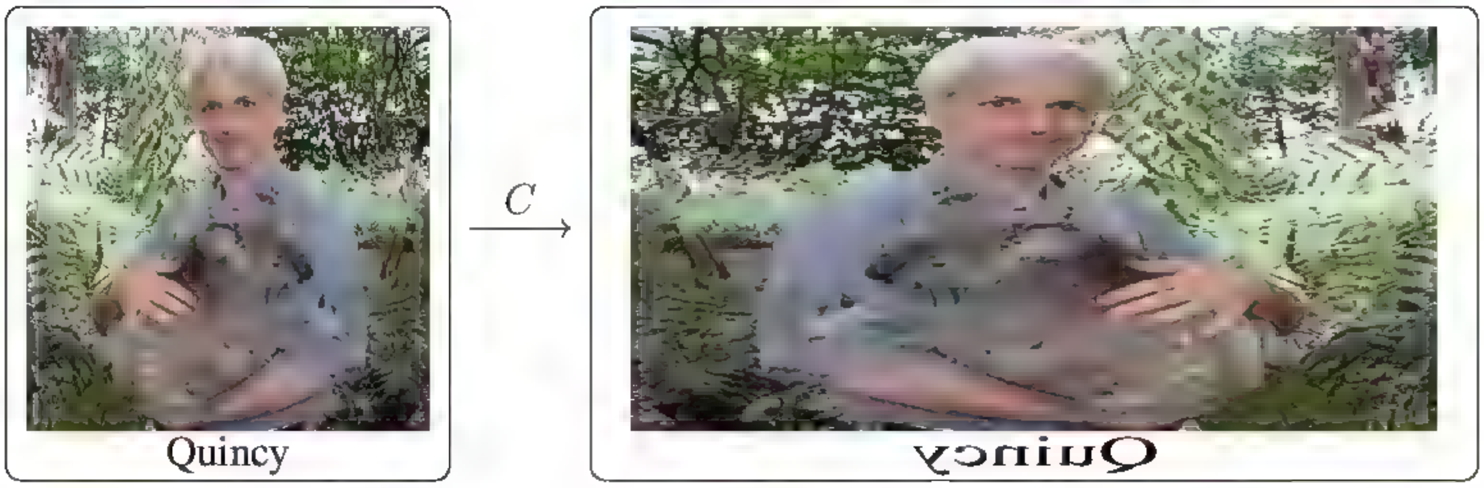


FIGURE E.3.5. The effect of C on the portrait of Quincy.

The distinction between B and C is that B is *orientation-preserving* while C is *orientation-reversing*. In general, an orientation-reversing transformation is one that switches what is "right" and what is "left." Note also that C *reverses* the direction of the text at the bottom of Quincy's portrait. This is another example of how C switches "right" and "left."

Let $D = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$. Geometrically, D represents rotation by 180° around the origin. Figure E.3.6 shows the result of applying T_D to the image of Quincy. Now the output image is upside down, but the right hand remains on the koala. Thus, D is orientation-preserving.

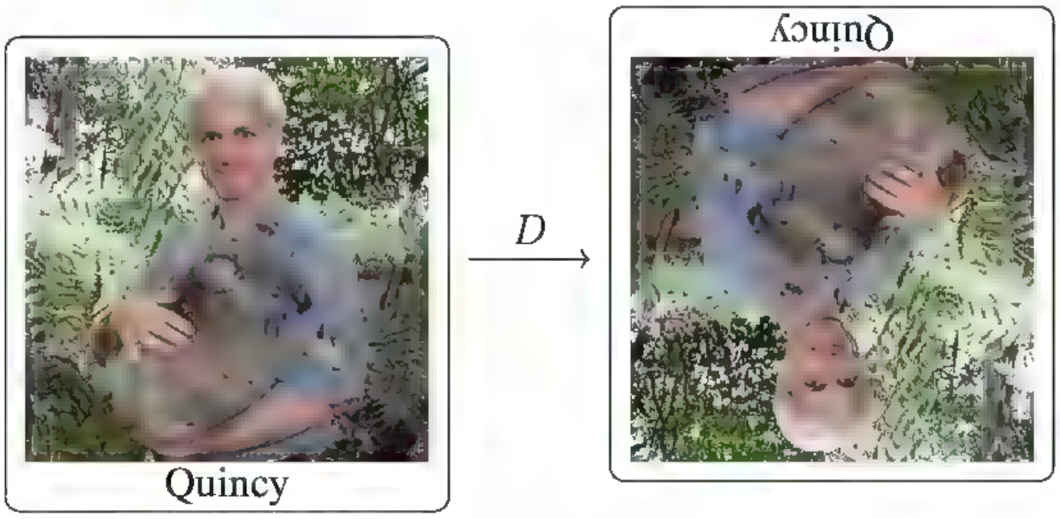


FIGURE E.3.6. The effect of D on the portrait of Quincy.

■

With the preceding example in mind, we can now describe the geometric meaning of the determinant.

For a 2×2 matrix A, $\det(A)$ is the "signed area magnification factor" of A. More precisely:

- If A is orientation-preserving then $\det(A)$ is the area-magnification factor.
- If A is orientation-reversing then $\det(A)$ is minus the area-magnification factor.

If A is an $n \times n$ matrix then $\det(A)$ is the *signed n-dimensional volume magnification factor* of A:

- If A is orientation-preserving then $\det(A)$ is the n-dimensional volume magnification factor.
- If A is orientation-reversing then $\det(A)$ is minus the n-dimensional volume magnification factor.

If the notions of orientation-preserving and orientation-reversing, or of n-dimensional volume (for $n > 3$), are unclear then don't worry: we haven't defined these concepts for general n. We hope the examples above give you a basic idea about these concepts in the 2-dimensional case. In practice, it is easy to tell whether a given matrix is orientation-preserving or orientation-reversing: just calculate the determinant and see whether it is positive or negative. Beware that the determinant might be 0; in that case, the concept of orientation doesn't apply and so the matrix is neither orientation-preserving nor orientation-reversing.

Example E.3.2. Let U be the ball of radius $r > 0$ centered at the origin in $\mathbf{R}^3$:

$$U = \{\mathbf{x} \in \mathbf{R}^3 : \|\mathbf{x}\| \leq r\}.$$

We shall now find the volume of the region $A(U)$ obtained from U by feeding points of U into T_A for

$$A = \begin{bmatrix} 1 & 1 & 3 \\ 0 & 2 & 7 \\ 2 & 2 & 3 \end{bmatrix}.$$

We have

$$\mathrm{vol}(A(U)) = |\det(A)|(\mathrm{vol}(U)) = |\det(A)| \cdot \frac{4}{3}\pi r^3,$$

so we just need to find $\det(A)$. This goes as follows:

$$\begin{vmatrix} 1 & 1 & 3 \\ 0 & 2 & 7 \\ 2 & 2 & 3 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 3 \\ 0 & 2 & 7 \\ 0 & 0 & -3 \end{vmatrix} \quad \text{(2(row 1) subtracted from row 3)}$$
$$= -6.$$

Thus

$$\mathrm{vol}(A(U)) = |-6| \cdot \frac{4}{3}\pi r^3 = 8\pi r^3.$$

■

In Remark 18.2.6 we gave a general interpretation of the determinant of a 2×2 matrix A in terms of area. The calculations there showed that $\det(A)$ equals plus or minus the area of the parallelogram whose edges are the columns of A, with sign "+" when A is orientation-preserving (i.e., the second column of A lies counterclockwise from the first column with an angle of at most 180°) and sign "−" if A is orientation-reversing. The n-dimensional analogue of a parallelogram is called a *parallelepiped*. (A parallelogram in $\mathbf{R}^2$ with a corner at $(0,0)$ and vectors $\mathbf{v}_1, \mathbf{v}_2$ among its edges consists of the points $x\mathbf{v}_1 + y\mathbf{v}_2 \in \mathbf{R}^2$ with $0 \leq x, y \leq 1$, shown in Figure 14.1.3 as the red region labeled "Q" there. For n vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbf{R}^n$ the parallelepiped P with a corner at $\mathbf{0}$ and the $\mathbf{v}_i$'s among its "edges" is defined to be the set of points $a_1\mathbf{v}_1 + \cdots + a_n\mathbf{v}_n \in \mathbf{R}^n$ with $0 \leq a_1, \ldots, a_n \leq 1$.) Arguing in a more sophisticated way, one can establish:

> Let A be an $n \times n$ matrix, and P_A the parallelepiped with a corner at the origin and the columns of A among its edges. If $\det(A) \neq 0$ then $\det(A)|$ equals the "n-dimensional volume" (appropriately defined when $n > 3$) of P_A and the determinant is positive when A is orientation-preserving and negative when A is orientation-reversing. If $\det(A) = 0$ then then P_A is contained in a hyperplane (due to Theorem E.4.5 below) and so its "n-dimensional volume" vanishes.

Example E.3.3. Consider the triangle PQR in $\mathbf{R}^2$ whose vertices are at the points $P = (1,2)$, $Q = (-2,3)$, and $R = (4,5)$. Here is a *general method* to compute its area by using a 2×2 determinant via the area interpretation discussed above (applied with $n = 2$).

Method: For a triangle T in $\mathbf{R}^2$, we apply displacement by the negative of a vertex to move T to a new triangle T' (so with the same area) having one vertex at the origin. The area of T' is half of the area of a parallelogram P whose area is $|\det(A)|$ for the 2×2 matrix A with columns given by the edge vectors of T' emanating from the origin as illustrated in Figure E.3.7 (also see Figure 18.2.2 in Remark 18.2.6).

FIGURE E.3.7. Triangle T moving by displacement to a triangle T' with a vertex at the origin, and T' is "half" of a parallelogram P.

We now apply this method to compute the area of the triangle PQR, by respectively moving each vertex to the origin. These three calculations will yield quite different 2×2 matrices A, but in all cases $|\det(A)|$ will be the same (namely, the area of the triangle).

First we use displacement by $-P$ to compute the area. Displacing by $-P = (-1, -2)$ yields T' with vertices at $-P + P = (0, 0)$, $-P + Q = (-3, 1)$, and $-P + R = (3, 3)$. Thus, T' is half of the parallelogram with a vertex at the origin and two edges along the vectors $\mathbf{v} = \begin{bmatrix} -3 \\ 1 \end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$ (so the other vertex of the parallelogram is $\mathbf{v} + \mathbf{w} = \begin{bmatrix} 0 \\ 4 \end{bmatrix}$). The area of the parallelogram is the absolute value of the determinant of the 2×2 matrix with columns given by $\mathbf{v}$ and $\mathbf{w}$. This determinant is

$$\det \begin{bmatrix} -3 & 3 \\ 1 & 3 \end{bmatrix} = -9 - 3 = -12,$$

so half of its absolute value is 6.

Next, we use displacement by $-Q$ to compute the area. Displacing by $-Q = (2, -3)$ yields T' with vertices at $-Q+P = (3, -1)$, $-Q+Q = (0, 0)$, and $-Q+R = (6, 2)$. Thus, T' is half of the parallelogram with a vertex at the origin and two edges along the vectors $\mathbf{v} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix} 6 \\ 2 \end{bmatrix}$ (so the other vertex of the parallelogram is $\mathbf{v} + \mathbf{w} = \begin{bmatrix} 9 \\ 1 \end{bmatrix}$). The area of the parallelogram is the absolute value of the determinant of the 2×2 matrix with columns given by $\mathbf{v}$ and $\mathbf{w}$. This determinant is

$$\det \begin{bmatrix} 3 & 6 \\ -1 & 2 \end{bmatrix} = 6 - (-6) = 12,$$

so half of its absolute value is 6, agreeing with the first area calculation (as it must).

Finally, we use displacement by $-R$ to compute the area. Displacing by $-R = (-4, -5)$ yields T' with vertices at $-R + P = (-3, -3)$, $-R + Q = (-6, -2)$, and $-R + R = (0, 0)$. Thus, T' is half of the parallelogram with a vertex at the origin and two edges along the vectors $\mathbf{v} = \begin{bmatrix} -3 \\ -3 \end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix} -6 \\ -2 \end{bmatrix}$ (so the other vertex of the parallelogram is $\mathbf{v} + \mathbf{w} = \begin{bmatrix} -9 \\ -5 \end{bmatrix}$). The area of the parallelogram is the absolute value of the determinant of the 2×2 matrix with columns given by $\mathbf{v}$ and $\mathbf{w}$. This determinant is

$$\det \begin{bmatrix} -3 & -6 \\ -3 & -2 \end{bmatrix} = 6 - 18 = -12,$$

so half of its absolute value is 6, agreeing with the two preceding area calculations (as it must). ■

E.4. Additional Properties of and Formulas for Determinants. In our study of matrix algebra, we encountered operations on $n \times n$ matrices such as matrix multiplication and transpose. It is useful to understand how these interact with the determinant.

Theorem E.4.1. For any $n \times n$ matrices A and B, we have $\det(AB) = \det(A)\det(B)$. If A is invertible then $\det(A) \neq 0$ and $\det(A^{-1}) = 1/\det(A)$.

Remark E.4.2. In Remark E.4.6 we will provide a refinement: the non-vanishing of $\det(A)$ is *equivalent* to the invertibility of A (rather than just being a consequence of the invertibility of A as asserted in Theorem E.4.1).

To understand why Theorem E.4.1 is true, let U be a region in $\mathbf{R}^n$ with volume equal to 1. Then

$$\mathrm{vol}((AB)(U)) = |\det(AB)|\mathrm{vol}(U) = |\det(AB)|.$$

On the other hand,

$$\begin{aligned}\mathrm{vol}((AB)(U)) &= \mathrm{vol}(A(B(U))) \\ &= |\det(A)|\mathrm{vol}(B(U)) \\ &= |\det(A)||\det(B)|\mathrm{vol}(U) \\ &= |\det(A)||\det(B)|.\end{aligned}$$

Thus $|\det(AB)| = |\det(A)|\,|\det(B)|$, so

$$\det(AB) = \pm\det(A)\det(B).$$

In fact, the orientations always work out so that

$$\det(AB) = \det(A)\det(B)$$

(see [Ap, Sec. 3.7] for a proof if you are interested). In the invertible case, if we take B to be A^{-1} then $AB = \mathrm{I}_n$, so

$$1 = \det(\mathrm{I}_n) = \det(AB) = \det(A)\det(B).$$

This implies that $\det(A)$ must be nonzero (as otherwise we would have $1 = 0\det(B) = 0$, which is absurd). Then we can divide by $\det(A)$ to get that $\det(A^{-1}) = \det(B) = 1/\det(A)$ as claimed.

Theorem E.4.3. For any $n \times n$ matrix A, $\det(A^\top) = \det(A)$.

The idea behind this result is that row operations on A correspond to column operations on $A^\top$ and vice-versa, with the *same effect* on each determinant (e.g., swapping two rows or two columns multiplies the determinant by -1, adding a multiple of one row or column to another has no effect, and multiplying a row or column by a scalar c multiplies the determinant by c).

Let's explain the idea in a 3×3 case before we take it up in general. Any matrix A can be changed into an upper triangular matrix by adding multiples of rows to other rows and possibly switching pairs of rows if necessary:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \rightsquigarrow \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ 0 & b_{22} & b_{23} \\ 0 & 0 & b_{33}. \end{bmatrix} = B.$$

(This was discussed in Remark E.1.5, where we saw that one can even arrange for B to be diagonal.) Thus, if there are k row swaps involved then

$$\det\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = (-1)^k \det\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ 0 & b_{22} & b_{23} \\ 0 & 0 & b_{33}. \end{bmatrix} = (-1)^k b_{11}b_{22}b_{33}. \tag{E.4.1}$$

Note that if we do *exactly* the same operations to the columns of $A^\top$ then we change it to the lower triangular matrix $B^\top$ (since applying column operations on the transpose $A^\top$ is "the same" as applying the corresponding row operations on A and then passing to the transpose):

$$A^\top = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} \rightsquigarrow \begin{bmatrix} b_{11} & 0 & 0 \\ b_{12} & b_{22} & 0 \\ b_{13} & b_{23} & b_{33} \end{bmatrix} = B^\top,$$

so

$$\det(A^\top) = (-1)^k b_{11}b_{22}b_{33}. \tag{E.4.2}$$

From (E.4.1) and (E.4.2), we see that $\det(A)$ and $\det(A^\top)$ are equal.

The general case goes in a similar way: row and column operations can always be used to bring a matrix to diagonal form, as we saw in Remark E.1.5. Hence, the problem of equality of $\det(A^\top)$ and $\det(A)$ can be transformed via compatible row and column operations on A and $A^\top$ into the case that A is diagonal. But in the diagonal case we have $A^\top = A$, so certainly $\det(A^\top) = \det(A)$!

Corollary E.4.4. If R is an $n \times n$ orthogonal matrix then $\det(R) = \pm 1$. In particular, if R is an orthogonal 2×2 or 3×3 matrix then its effect on $\mathbf{R}^2$ or $\mathbf{R}^3$ respectively is orientation-preserving precisely when $\det(R) = 1$.

PROOF. Applying the determinant to both sides of the orthogonality equation $RR^\top = \mathrm{I}_n$ yields (with the help of Theorems E.4.1 and E.4.3)

$$1 = \det(\mathrm{I}_n) = \det(RR^\top) = \det(R)\det(R^\top) = \det(R)\det(R) = \det(R)^2,$$

so $\det(R) = \pm 1$. The sign of the determinant keeps track of whether or not a linear motion of $\mathbf{R}^2$ or $\mathbf{R}^3$ preserves the sense of "orientation" (in effect, the distinction between left-handedness and right-handedness, which is flipped if one looks in a mirror), so when $n = 2, 3$ we have that R is orientation-preserving precisely when $\det(R) > 0$, or equivalently $\det(R) = 1$. ☐

Determinants also encode linear dependence for any n vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ in $\mathbf{R}^n$:

Theorem E.4.5. For an $n \times n$ matrix A, $\det(A) = 0$ precisely when the n columns $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbf{R}^n$ of A are linearly dependent. Put another way, the n vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbf{R}^n$ are linearly independent (or equivalently, constitute a basis of $\mathbf{R}^n$) precisely when $\det(A) \neq 0$.

Likewise, $\det(A) = 0$ precisely when the n rows of A are linearly dependent, and $\det(A) \neq 0$ precisely when the n rows of A are linearly independent.

To explain this result, first note that the result for rows of A is the same as the result for columns of $A^\top$, and by Theorem E.4.3 the matrices A and $A^\top$ have the *same* determinant. Hence, it is enough to check that $\det(A) = 0$ exactly when the columns $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are linearly dependent in $\mathbf{R}^n$.

As a warm-up to the general case, let's first suppose $n = 3$ and explain why linear dependence of the columns $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$ of a 3×3 matrix forces the vanishing of the determinant. By linear dependence, one of these columns is a linear combination of the other two. Suppose, for example, than $\mathbf{w}$ is a linear combination of $\mathbf{u}$ and $\mathbf{v}$:

$$\mathbf{w} = a\mathbf{u} + b\mathbf{v}$$

for some scalars a and b. Then subtracting a times column 1 from column 3 and then subtracting b times column 2 from column 3 does not change the determinant but it makes the third column vanish:

$$\det(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \det(\mathbf{u}, \mathbf{v}, \mathbf{w} - a\mathbf{u} - b\mathbf{v}) = \det(\mathbf{u}, \mathbf{v}, \mathbf{0}).$$

The right side vanishes by Theorem E.1.4.

In the general case, linear dependence says $\sum c_j\mathbf{v}_j = \mathbf{0}$ for some scalars $c_1, \dots, c_n \in \mathbf{R}$ that aren't all 0. Some c_{j_0} is nonzero, so if we divide through by c_{j_0} then

$$\mathbf{v}_{j_0} = \sum_{j \neq j_0} -(c_j/c_{j_0})\mathbf{v}_j.$$

The determinant is unaffected by adding multiples of one column to another, so if we add to $\mathbf{v}_{j_0}$ the multiple by c_j/c_{j_0} of the jth column $\mathbf{v}_j$ for every $j \neq j_0$ then $\det(A)$ doesn't change but the effect on the j_0th column is to turn it into

$$\begin{aligned}\mathbf{v}_{j_0} + \sum_{j \neq j_0}(c_j/c_{j_0})\mathbf{v}_j &= \sum_{j \neq j_0} -(c_j/c_{j_0})\mathbf{v}_j + \sum_{j \neq j_0}(c_j/c_{j_0})\mathbf{v}_j \\ &= \sum_{j \neq j_0}(-(c_j/c_{j_0}) + c_j/c_{j_0})\mathbf{v}_j \\ &= \sum_{j \neq j_0} 0\,\mathbf{v}_j \\ &= \mathbf{0}.\end{aligned}$$

We have arrived at an entire column of 0's, so $\det(A) = 0$ (by Theorem E.1.4).

Now going in reverse, suppose $\det(A) = 0$. We want to show that $\mathbf{v}_1, \dots, \mathbf{v}_n$ is linear dependent. *It is hard to establish this directly*, so instead we are going to proceed by contradiction: we assume the collection $\mathbf{v}_1, \dots, \mathbf{v}_n$ is linearly *independent* and will deduce a contradiction (so the assumption of linear independence fails, and hence linear dependence must hold, as desired). The key point is that by assuming linear independence we can apply a serious ingredient: general results about dimension and bases.

Since $\dim \mathbf{R}^n = n$, any n linearly independent vectors in $\mathbf{R}^n$ constitute a basis and in particular span $\mathbf{R}^n$. Hence, every vector $\mathbf{b} \in \mathbf{R}^n$ is a linear combination of the $\mathbf{v}_j$'s. In particular, each of the standard basis vectors $\mathbf{e}_1, \dots, \mathbf{e}_n \in \mathbf{R}^n$ is in the span of the $\mathbf{v}_j$'s. We can write this as a collection of equations

$$\mathbf{e}_j = \sum_{i=1}^{n} c_{ij}\mathbf{v}_i$$

for some scalars c_{ij}. If we let C be the $n \times n$ matrix (c_{ij}) then the product matrix AC has first column

$$\sum_{i=1}^{n} c_{i1}\mathbf{v}_i = \mathbf{e}_1$$

and likewise jth column $\sum_{i=1}^{n} c_{ij}\mathbf{v}_i = \mathbf{e}_j$ for all $1 \leq j \leq n$.

This says that AC is the $n \times n$ matrix whose jth column is $\mathbf{e}_j$, and that is exactly the identity matrix I_n. In other words, when the columns of A are linearly independent we have built an $n \times n$ matrix C so that $AC = \mathrm{I}_n$. Now we apply Theorem E.4.1:

$$1 = \det(\mathrm{I}_n) = \det(AC) = \det(A)\det(C).$$

But we assumed $\det(A) = 0$, so the right side is $0\det(C) = 0$, yielding $1 = 0$, an absurdity. This completes our verification of Theorem E.4.5!

Remark E.4.6. A consequence of Theorem E.4.5 is that $\det(A) \neq 0$ exactly when A is invertible. Indeed, in Theorem E.4.1 we saw that if A has an inverse then $\det(A)$ cannot vanish. In the other direction, if $\det(A) \neq 0$ then by Theorem E.4.5 the n columns in $\mathbf{R}^n$ are linearly independent, so the argument in the second part of the explanation for Theorem E.4.5 produces an $n \times n$ matrix C so that $AC = \mathrm{I}_n$, so A is invertible.

Here is a formula that computes an $n \times n$ determinant in terms of an $(n-1) \times (n-1)$ determinant in a special case, which we will then build upon to give a general procedure to compute determinants.

Proposition E.4.7. For $n > 1$, if all entries in the ith row of an $n \times n$ matrix are 0 except for possibly the ij-entry then

$$\det(A) = (-1)^{i+j} a_{ij} \det(\widetilde{A}_{ij}),$$

where $\widetilde{A}_{ij}$ is the $(n-1) \times (n-1)$ matrix obtained by removing row i and column j from A.

Before we discuss why Proposition E.4.7 is true, we illustrate it with some examples in order to convey what it is saying.

Example E.4.8. The matrix

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 4 \\ 5 & 6 & 7 \end{bmatrix}$$

has all entries in the second row vanishing except for the entry in the 23 position, so

$$\begin{vmatrix} 1 & 2 & 3 \\ 0 & 0 & 4 \\ 5 & 6 & 7 \end{vmatrix} = (-1)^{2+3} 4 \begin{vmatrix} 1 & 2 \\ 5 & 6 \end{vmatrix} = -4(1 \cdot 6 - 2 \cdot 5) = -4(-4) = 16.$$

Likewise,

$$\begin{vmatrix} a & b & c & d \\ e & f & 0 & g \\ h & i & 0 & j \\ k & \ell & 0 & m \end{vmatrix} = (-1)^{1+3} c \begin{vmatrix} e & f & g \\ h & i & j \\ k & \ell & m \end{vmatrix}.$$

■

Example E.4.9. Proposition E.4.7 can be made applicable by first using row or column operations to *create* many 0's as we did in Remark E.1.5. For instance, let us compute the determinant of

$$\begin{bmatrix} 1 & 2 & 5 \\ 0 & 1 & 1 \\ 5 & 4 & 6 \end{bmatrix}.$$

To make more 0's along the second row we can subtract column 2 from column 3 to get a matrix whose second row has only one nonzero entry (so we can use Proposition E.4.7):

$$\begin{aligned} \begin{vmatrix} 1 & 2 & 5 \\ 0 & 1 & 1 \\ 5 & 4 & 6 \end{vmatrix} = \begin{vmatrix} 1 & 2 & 3 \\ 0 & 1 & 0 \\ 5 & 4 & 2 \end{vmatrix} &= (-1)^{2+2} 1 \begin{vmatrix} 1 & 3 \\ 5 & 2 \end{vmatrix} \\ &= 1(1 \cdot 2 - 5 \cdot 3) \\ &= -13. \end{aligned}$$

We could have proceeded in an entirely different way, using row operations instead of column operations, to still arrive at the same answer (assuming we avoid arithmetic errors): to introduce more 0's along the first row (rather than along the second column) we can subtract 5 times row 1 from row 3 (and then use Proposition E.4.7 on column 1):

$$\begin{aligned}\begin{vmatrix}1 & 2 & 5\\ 0 & 1 & 1\\ 5 & 4 & 6\end{vmatrix} &= \begin{vmatrix}1 & 2 & 5\\ 0 & 1 & 1\\ 0 & -6 & -19\end{vmatrix}\\ &= (-1)^{1+1}\,1\begin{vmatrix}1 & 1\\ -6 & -19\end{vmatrix}\\ &= (-1)^2(1(-19) - 1(-6))\\ &= -19+6\\ &= -13.\end{aligned}$$

And we can also calculate the determinant without using Proposition E.4.7 at all (so there is really a lot of flexibility in how to compute determinants):

$$\begin{aligned}\begin{vmatrix}1 & 2 & 5\\ 0 & 1 & 1\\ 5 & 4 & 6\end{vmatrix} &= \begin{vmatrix}1 & 2 & 5\\ 0 & 1 & 1\\ 0 & -6 & -19\end{vmatrix} && \text{(5(row 1) subtracted from row 3)}\\ &= \begin{vmatrix}1 & 2 & 5\\ 0 & 1 & 1\\ 0 & 0 & -13\end{vmatrix} && \text{(6 (row 2) added to row 3)}\\ &= (1)(1)(-13)\\ &= -13.\end{aligned}$$

■

The explanation for Proposition E.4.7 rests on using swaps of rows and columns to pass to the case $i = j = 1$ as follows. Moving the ith row up one step at a time by swapping it repeatedly with the row just above it involves $i-1$ row swaps and brings the original ith row to the top of the matrix. Now the first row vanishes except for possibly the jth entry. In the same way, using $j-1$ successive column swaps to the left brings the jth column all the way to the left. The outcome B of this process is related to the original matrix A as follows:

- The original entry a_{ij} in A is now in the upper left of B: $b_{11} = a_{ij}$.
- The $(n-1)\times(n-1)$ matrix $\widetilde{A}_{ij}$ obtained by deleting the ith row and jth column from the original $n \times n$ matrix is now exactly the lower-right $(n-1)\times(n-1)$ matrix obtained by deleting the first row and column from B; that is, $\widetilde{B}_{11} = \widetilde{A}_{ij}$. (This is the reason that we carried out the row and column swaps using successive *adjacent* rows and columns, so $\widetilde{B}_{11}$ and $\widetilde{A}_{ij}$ are exactly the same, and not off from each other by some row or column swaps.)

Hence, $\det(A) = (-1)^{(i-1)+(j-1)}\det(B) = (-1)^{i+j-2}\det(B) = (-1)^{i+j}\det(B)$, so if we can establish the result for B (the case $i = j = 1$) then

$$\det(A) = (-1)^{i+j} b_{11}\det(\widetilde{B}_{11}) = (-1)^{i+j} a_{ij}\det(\widetilde{A}_{ij}),$$

which is what we want. So by using row and column swaps, we have brought ourselves to the case $i = j = 1$; i.e., we can rename B as A.

We now want to show

$$\det(A) \overset{?}{=} a_{11}\det(\widetilde{A}_{11})$$

when the entire first row vanishes away from the left entry. If $a_{11} = 0$ then we have an entire row of 0's, so $\det(A) = 0$ by Theorem E.1.4 and the desired formula says $0 \overset{?}{=} 0 \det(\widetilde{A}_{11})$, which is certainly true (regardless of the value of $\det(\widetilde{A}_{11})$).

Suppose instead $a_{11} \neq 0$, so we can use the nonzero a_{11} as a pivot to clear out the entire first column as well (using row operations as in Remark E.1.5 to subtract off a suitable multiple of the first row from each of the other rows). Now consider using row or column operations on the $(n-1) \times (n-1)$ submatrix $\widetilde{A}_{11}$ to turn that into a diagonal matrix (as can be done, by Remark E.1.5). This has *no impact* on the first row or the first column of A (as those are now 0's away from the upper-left corner). Moreover, the effect of such row and column operations on both sides of the desired formula $\det(A) \overset{?}{=} a_{11} \det(\widetilde{A}_{11})$ (essentially to change by possibly a sign) is *the same*, so the validity or not of the desired formula is reduced to the case that $\widetilde{A}_{11}$ is diagonal. But then A is diagonal, so now

$$\det(A) = a_{11}a_{22} \cdots a_{nn} = a_{11} \det(\widetilde{A}_{11})$$

(because $\widetilde{A}_{11}$ is diagonal with entries $a_{22}, \ldots, a_{nn}$), so we are done!

Although Proposition E.4.7 might look too special to be of much use in general, we can always leverage the linearity properties of determinants to put ourselves into a situation where the hypotheses of Proposition E.4.7 really hold. To see this, it is simplest first to explain the idea in the case $n = 3$: we claim the general formula

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} a & c \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix}. \tag{E.4.3}$$

(Informally, this is called "expanding along the first row" since we move across the first row and multiply each entry against the 2×2 determinant of what remains when we remove the row and column containing the chosen entry. We also have to use alternating signs when we put it all together.)

The key insight to get the formula (E.4.3) is to express the first row as a linear combination of row vectors which are each 0 away from one entry:

$$\begin{bmatrix} a & b & c \end{bmatrix} = \begin{bmatrix} a & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & b & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & c \end{bmatrix}.$$

Thus, we can use linearity of determinants in the first row along with Proposition E.4.7 (for the case $i = 1$) to get

$$\begin{aligned} \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} &= \begin{vmatrix} a & 0 & 0 \\ d & e & f \\ g & h & i \end{vmatrix} + \begin{vmatrix} 0 & b & 0 \\ d & e & f \\ g & h & i \end{vmatrix} + \begin{vmatrix} 0 & 0 & c \\ d & e & f \\ g & h & i \end{vmatrix} \\ &= (-1)^{1+1} a \begin{vmatrix} e & f \\ h & i \end{vmatrix} + (-1)^{1+2} b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + (-1)^{1+3} c \begin{vmatrix} d & e \\ g & h \end{vmatrix} \\ &= a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix}. \end{aligned}$$

The same reasoning for an $n \times n$ matrix gives:

$$\begin{aligned} \det(A) &= a_{11} \det \widetilde{A}_{11} - a_{12} \det \widetilde{A}_{12} + a_{13} \det \widetilde{A}_{13} - \cdots \\ &= \sum_{j=1}^{n} (-1)^{1+j} a_{1j} \det(\widetilde{A}_{1j}). \end{aligned}$$

This is called *expanding* $\det(A)$ *along row* 1. More generally, one can expand along row i for any i to get:

Theorem E.4.10. For any $1 \le i \le n$,

$$\det(A) = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} \det(\widetilde{A}_{ij}).$$

Since $\det(A) = \det(A^\top)$ (Theorem E.4.3), one can also expand along column j (for any j):

$$\det A = \sum_{i=1}^{n} (-1)^{i+j} a_{ij} \det(\widetilde{A}_{ij}).$$

For example, we can expand along column 3 to get

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = (-1)^{1+3} a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} + (-1)^{2+3} a_{23} \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} + (-1)^{3+3} a_{33} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

$$= a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} - a_{23} \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} + a_{33} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

and expand along column 2 to get

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = (-1)^{1+2} a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + (-1)^{2+2} a_{22} \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} + (-1)^{3+2} a_{32} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix}$$

$$= -a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{22} \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} - a_{32} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix}.$$

Observe that the alternating signs in the expansions along column 3 and column 2 are not the same: for column 3 we have "$+ - +$" and for column 2 we have "$- + -$". The trick to remembering what signs to use when expanding along a row or column in Theorem E.4.10 is to think about the $n \times n$ array of "+" and "−" alternating horizontally and vertically beginning with "+" in the upper left, as in the following for the cases $n = 2, 3, 4$ respectively:

$$\begin{array}{cc} + & - \\ - & + \end{array} \qquad\qquad \begin{array}{ccc} + & - & + \\ - & + & - \\ + & - & + \end{array} \qquad\qquad \begin{array}{cccc} + & - & + & - \\ - & + & - & + \\ + & - & + & - \\ - & + & - & + \end{array}$$

(In each of these arrays, the ij-entry is the sign of $(-1)^{i+j}$ for $1 \le i, j \le n$.)

Expanding along a row or along a column is usually *not* an efficient way to calculate determinants (so its significance is more for theoretical purposes rather than for numerical computations). It is better to use row and/or column operations to get to a triangular form, or to get a row (or column) all but one of whose entries is 0 so that one can apply the simpler Proposition E.4.7 (as in Example E.4.8).

Remark E.4.11. For an $n \times n$ matrix A that is invertible (equivalently, $\det(A) \neq 0$), here is an explicit (but computationally inefficient) formula for A^{-1}: the ij-entry of A^{-1} is $(-1)^{i+j} \det(\widetilde{A}_{ji}) / \det(A)$, where $\widetilde{A}_{ji}$ is the $(n-1) \times (n-1)$ matrix obtained by removing row j and column i from A. For a proof, see [**Ap**, Thm. 3.12].

E.5. Applications to eigenvalues and interpolation. An instance of using the vanishing of a determinant to detect non-invertibility of a square matrix is that it gives a tool to detect eigenvalues, as follows. In Theorem 23.3.1 we saw that for a 2×2 matrix, its eigenvalues are exactly the roots of an associated quadratic polynomial. The version for a $n \times n$ matrix A with any $n \ge 1$ is:

Theorem E.5.1. A scalar λ is an eigenvalue of A *exactly when* $\det(\lambda \mathrm{I}_n - A) = 0$.

This determinant is a degree-n polynomial in λ (see Remark E.5.2); it is called the *characteristic polynomial* of A. For $n = 2$ this recovers the result in Theorem 23.3.1.

To explain Theorem E.5.1, first note that by Theorem E.4.5 this vanishing is exactly the condition that the columns of $\lambda \mathrm{I}_n - A$ are linearly dependent, and that in turn amounts to the existence of a *nonzero* $\mathbf{v} \in \mathbf{R}^n$ for which

$$(\lambda \mathrm{I}_n - A)\mathbf{v} = \mathbf{0}$$

(since a matrix-vector product $M\mathbf{x}$ is the linear combination of the columns of M for which the ith column of M is multiplied against the ith entry x_i in $\mathbf{x}$, by Theorem 13.4.1, so $M\mathbf{x} = \mathbf{0}$ with $\mathbf{x}$ nonzero precisely expresses linear dependence for the columns of M by Theorem 19.1.5). But this is exactly the condition $\lambda\, \mathrm{I}_n\mathbf{v} - A\mathbf{v} = \mathbf{0}$, which is to say

$$A\mathbf{v} = \lambda \mathbf{v},$$

the very meaning of the nonzero $\mathbf{v}$ being an eigenvector of A with eigenvalue λ.

Remark E.5.2. As an application of Theorem E.4.10, we can show that the determinant $\det(\lambda \mathrm{I}_n - A)$ whose vanishing characterizes the eigenvalues of A in Theorem E.5.1 is a degree-n polynomial in λ. Explaining this for general n gets bogged down in notation, but already in the case $n = 3$ one can see exactly what is going on, so here is the calculation for $n = 3$: by expanding along the first row, the determinant

$$\det(\lambda \mathrm{I}_3 - A) = \begin{vmatrix} \lambda - a_{11} & -a_{12} & -a_{13} \\ -a_{21} & \lambda - a_{22} & -a_{23} \\ -a_{31} & -a_{32} & \lambda - a_{33} \end{vmatrix}$$

is equal to

$$(\lambda - a_{11}) \begin{vmatrix} \lambda - a_{22} & -a_{23} \\ -a_{32} & \lambda - a_{33} \end{vmatrix} - (-a_{12}) \begin{vmatrix} -a_{21} & -a_{23} \\ -a_{31} & \lambda - a_{33} \end{vmatrix} + (-a_{13}) \begin{vmatrix} -a_{21} & \lambda - a_{22} \\ -a_{31} & -a_{32} \end{vmatrix}.$$

This may look like a mess, but there are two key observations:

- the first term is $\lambda - a_{11}$ times an 2×2 version of the same kind of expression, which we already know is a quadratic polynomial (it is exactly what is given in Theorem 23.3.1), so altogether the first term is a cubic polynomial in λ,
- the other terms involve 2×2 determinants that we can expand along their first column to get polynomial expressions in λ with degree at most 1.

Hence, when we put everything together, the first term is the only one contributing in degree 3: the others contribute in degree at most 1. The overall result is therefore a cubic polynomial. In the general case one shows that the result for $(n-1) \times (n-1)$ matrices (as in the above calculations for $n = 3$ where we used the case of 2×2 matrices from Theorem 23.3.1) and a bit of algebra yields the result for $n \times n$ matrices.

Remark E.5.3. As an application of the case $n = 3$ in Remark E.5.2, we can now explain a fact which was mentioned in Remark 20.7.3: any (orientation-preserving) rigid motion of $\mathbf{R}^3$ that preserves the origin must be rotation around some line through the origin by some angle. By Theorem 20.4.1, any rigid motion of $\mathbf{R}^3$ preserving the origin arises from an orthogonal 3×3 matrix R. By Corollary E.4.4 we have $\det(R) = 1$. We need to show any such R is rotation by some angle around a line through the origin.

Now eigenvalues assist us in a crucial way: although a 2×2 matrix can fail to have an eigenvalue (e.g., rotation through an angle $0 < \theta < \pi$), *every* 3×3 matrix has an eigenvalue $\lambda \in \mathbf{R}$! Indeed, the eigenvalues are the roots of the characteristic polynomial, and for a 3×3 matrix this is a cubic polynomial of the form $f(x) = x^3 + ax^2 + bx + c$. But every cubic polynomial with leading term x^3 and coefficients in $\mathbf{R}$ has a root in $\mathbf{R}$! This is a consequence of the Intermediate Value Theorem because the graph of such a cubic polynomial (unlike a quadratic polynomial) shoots off both arbitrarily high up positive and arbitrarily low down negative (think about the graph of $y = x^3$, for instance). To be precise, since x^3 dominates the other terms ax^2, bx, and c by a lot in absolute value when $|x|$ is large, $f(x)$ is very large for large x and is very negative for very negative x. By the Intermediate Value Theorem from single-variable calculus, therefore there must be a zero of f between those extremes (positive and negative values of f). Such a zero is exactly an eigenvalue.

So R has an eigenvalue λ for some eigenvector $\mathbf{v}$. The effect of R on the line $\ell = \mathrm{span}(\mathbf{v})$ is multiplication by λ. Hence, the effect of R on this line is to alter length by the factor $|\lambda|$, so $|\lambda| = 1$ since R is length-preserving. That is, $\lambda = \pm 1$. If $\lambda = 1$ then ℓ is a line on which the effect of R is to do nothing. We claim that in such cases R is a rotation around the axis ℓ. Since R is angle-preserving, it preserves the property of being orthogonal to ℓ. Hence, for the plane $\ell^{\perp}$ through $\mathbf{0}$ perpendicular to ℓ, R carries $\ell^{\perp}$ into itself. The resulting effect of R on that plane is linear and length-preserving, hence also a "rigid motion", and an argument with determinants shows that it is also orientation-preserving on $\ell^{\perp}$ because R is orientation-preserving on $\mathbf{R}^3$ and on ℓ (where it does nothing). Consequently (please convince yourself of this), on $\ell^{\perp}$ the effect of R *must* be rotation through some angle θ. Since every 3-vector is a sum of a vector in ℓ and a vector in the perpendicular plane $\ell^{\perp}$ (Theorem 6.2.4), by applying R to such a sum we see that (when $\lambda = 1$) R is exactly the rotation around ℓ by the angle θ!

What if $\lambda = -1$? In such cases the linear effect of R on the plane $\ell^{\perp}$ is certainly orthogonal (since R is orthogonal on $\mathbf{R}^3$) and it is *orientation-reversing* (by an argument using multiplicativity of determinants and the fact that R is orientation-preserving on $\mathbf{R}^3$ but orientation-reversing on the line ℓ perpendicular to that plane, as R acts on ℓ through multiplication by $\lambda = -1$). Hence, the effect of R on the plane $\ell^{\perp}$ has determinant -1. In particular, if we choose a basis of this plane to express the linear effect of R on it in terms of a 2×2 matrix B, the quadratic characteristic polynomial of B has constant term $\det(B) = -1$. By the quadratic formula, any quadratic polynomial of the form $x^2 + ax - 1$ has two different roots in $\mathbf{R}$ because the discriminant $a^2 - 4(-1) = a^2 + 4$ is always positive (regardless of the value of a). Such different roots λ', λ'' for the characteristic polynomial of B are different eigenvalues for B, and corresponding eigenvectors are nonzero vectors $\mathbf{v}', \mathbf{v}'' \in \ell^{\perp}$ satisfying $R\mathbf{v}' = \lambda'\mathbf{v}'$ and $R\mathbf{v}'' = \lambda''\mathbf{v}''$. Thus, λ' and λ'' are different eigenvalues of R. But we have already seen that the only *possible* eigenvalues of R are ± 1, so one of λ' or λ'' must be 1! So even when $\lambda = -1$, we found that 1 still occurs as an eigenvalue for R (along another line, located in the plane $\ell^{\perp}$). Hence, we can run the earlier "$\lambda = 1$" argument using that line in place of ℓ to get the desired conclusion.

In Theorem E.4.5 we saw that the linear independence of n vectors in $\mathbf{R}^n$ is characterized by the nonvanishing of the determinant of the $n \times n$ matrix whose columns are those vectors. This has an interesting application to higher-degree curve fitting to a collection of points, also called *polynomial interpolation*. The classic result on polynomial interpolation is:

Theorem E.5.4 (Lagrange Interpolation). Given $d + 1$ different numbers $x_1, \dots, x_{d+1}$ and $d + 1$ values $y_1, \dots, y_{d+1}$, there is exactly one polynomial $f(x) = a_0 + a_1x + \cdots + a_dx^d$ of degree at most d for which

$$f(x_1) = y_1, f(x_2) = y_2, \dots, f(x_{d+1}) = y_{d+1}.$$

In other words, there is exactly one such f whose graph passes through $(x_1, y_1), (x_2, y_2), \dots, (x_{d+1}, y_{d+1})$.

Note that we do *not* make any assumption of distinctness among the y_j's (only for the x_i's). In the case $d = 1$, the result holds because there is exactly one line through 2 different points. For many applications, one needs to go further and design efficient algorithms to construct f from the given data. There are a variety of ways to do this (an explicit formula for f is due independently to Waring and Lagrange, though it isn't so useful on a computer when d is large), and here we want to discuss the role of a remarkable determinantal identity in its proof:

Theorem E.5.5 (Vandermonde's determinant). For any numbers $x_1, \ldots, x_{d+1}$, we have

$$\begin{vmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^d \\ 1 & x_2 & x_2^2 & \cdots & x_2^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{d+1} & x_{d+1}^2 & \cdots & x_{d+1}^d \end{vmatrix} = \prod_{1 \le i < j \le d+1} (x_j - x_i),$$

so if the x_i's are pairwise different then this determinant is nonzero and hence the matrix is invertible.

(In this result, the notation $\prod$ – capital Greek "P" – is shorthand for "product" in the same way that $\sum$ – capital Greek "S" – is shorthand for "sum".)

As a special case of Theorem E.5.5, if $d = 2$ then the result is asserting the formula

$$\begin{vmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{vmatrix} = (x_2 - x_1)(x_3 - x_1)(x_3 - x_2).$$

For those who are interested, a short proof of the general case of Theorem E.5.5 via row and column operations (using induction on d) is given on the Wikipedia page for "Vandermonde matrix" (the name for the matrix whose determinant is being considered in Theorem E.5.5).

The Vandermonde matrix is used in algorithms that perform rapid evaluation of a given polynomial at many points, and is *very* useful for the "inverse problem" of polynomial interpolation. More specifically, the conditions $f(x_i) = y_i$ in Theorem E.5.4 amount to a system of equations

$$\begin{aligned} a_0 + a_1 x_1 + \cdots + a_d x_1^d &= y_1 \\ a_0 + a_1 x_2 + \cdots + a_d x_2^d &= y_2 \\ &\vdots \\ a_0 + a_1 x_d + \cdots + a_d x_d^d &= y_d \end{aligned}$$

which can be written in vector form as

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^d \\ 1 & x_2 & x_2^2 & \cdots & x_2^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{d+1} & x_{d+1}^2 & \cdots & x_{d+1}^d \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_d \end{bmatrix}.$$

Since the x_i's are pairwise different, so the $(d+1) \times (d+1)$ matrix A on the left is invertible by Theorem E.5.5, we can solve for the a_i's in terms of the y_j's, namely $\mathbf{a} = A^{-1}\mathbf{y}$. This establishes Theorem E.5.4.

Remark E.5.6. The problem of polynomial interpolation (i.e., finding a polynomial of controlled degree whose graph passes through a given finite collection of points) arises in a variety of situations. For instance, it occurs in cryptography for secure multi-party computation (see [**DKMSZ**]) and Shamir secret-sharing (see [**Sham**]), and in coding theory via the Reed–Solomon error-correcting codes (used in applications as varied as supermarket scanners, QR code readers, and deep space communications).

If we regard $x_1, \ldots, x_{d+1}$ as fixed and $y_1, \ldots, y_{d+1}$ as varying, it makes sense to ask how the coefficients $a_0, \ldots, a_d$ of the unique f in Theorem E.5.4 depend on the values y_i being interpolated. The formula $\mathbf{a} = A^{-1}\mathbf{y}$ shows that this dependence is *linear*, a fact that is *very* useful to know. (In practice there are faster ways to compute the a_j's than to use the recipe $A^{-1}\mathbf{y}$ inverting the Vandermonde matrix, but knowing the a_j's depend linearly on the y_i's is useful; this linearity can also be seen in other ways.)

A somewhat different application of the Vandermonde determinant formula arises in higher-degree curve-fitting ("approximate polynomial interpolation") as follows. In Section 20.6 we gave a matrix-algebra method to find the quadratic polynomial $ax^2 + bx + c$ whose graph (among those of all quadratic polynomials) best fits n given data points (x_i, y_i). The method required knowing that as long as there are at least 3 different values among the x_i's (a reasonable assumption, since otherwise the data points would lie on one or two vertical lines, so it wouldn't make any sense to seek a quadratic function fitting the data) then the n-vectors

$$\mathbf{1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{X}' = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \vdots \\ x_n^2 \end{bmatrix}$$

are linearly independent. (The method we used in Section 20.6 involved a projection into the span of those three vectors, and we needed to know that these vectors are a *basis* of their span to ensure the invertibility of the 3×3 matrix $\mathbf{V}^\top \mathbf{V}$, where $\mathbf{V}$ is the $n \times 3$ matrix whose columns are those same three vectors.)

To understand why such linear independence holds when there are at least 3 different values for the x_i's, it is convenient to consider a more general linear independence claim: if at least $d+1$ of the x_i's are different from each other then we claim that the n-vectors

$$\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \quad \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \begin{bmatrix} x_1^2 \\ x_2^2 \\ \vdots \\ x_n^2 \end{bmatrix}, \quad \dots, \quad \begin{bmatrix} x_1^d \\ x_2^d \\ \vdots \\ x_n^d \end{bmatrix}$$

are linearly independent; note that there are $d+1$ such vectors. (For $d = 2$ this recovers the situation that arose in the quadratic-fit example in Section 20.6.) Since $d+1$ of the x_i's are assumed to be pairwise different, we can rearrange the x_i's so that that $x_1, \dots, x_{d+1}$ are pairwise different (matching the setup in Theorem E.5.5, except that now we also have $x_{d+2}, \dots, x_n$).

It isn't the columns of the $(d+1) \times (d+1)$ matrix in Theorem E.5.5 that arise, but rather the columns of the $n \times (d+1)$ matrix

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^d \\ 1 & x_2 & x_2^2 & \dots & x_2^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^d \end{bmatrix}. \tag{E.5.1}$$

Theorem E.5.5 ensures that the top $(d+1) \times (d+1)$ submatrix of (E.5.1) is invertible, so the columns of that $(d+1) \times (d+1)$ submatrix are linearly independent (by Theorem E.4.5). Thus, it suffices to show that the linear independence of those $d+1$ vectors in $\mathbf{R}^{d+1}$ implies the linear independence of the $d+1$ columns in $\mathbf{R}^n$ that appear in (E.5.1).

By writing $\mathbf{v}_1, \dots, \mathbf{v}_{d+1} \in \mathbf{R}^n$ for the columns in (E.5.1), to show they are linearly independent it is the same (by Theorem 19.1.5) to show that the only way we can have a vanishing linear combination

$$c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + \cdots + c_{d+1} \mathbf{v}_{d+1} = \mathbf{0} \tag{E.5.2}$$

in $\mathbf{R}^n$ is when the coefficients c_j all vanish. But such vanishing in $\mathbf{R}^n$ implies the corresponding vanishing in $\mathbf{R}^{d+1}$ upon focusing on the first $d+1$ entries in these vectors. In other words, if $\mathbf{v}'_j \in \mathbf{R}^{d+1}$ is obtained from $\mathbf{v}_j$ by chopping off everything after the $(d+1)$th entry then certainly

$$c_1 \mathbf{v}'_1 + c_2 \mathbf{v}'_2 + \cdots + c_{d+1} \mathbf{v}'_{d+1} = \mathbf{0} \tag{E.5.3}$$

in $\mathbf{R}^{d+1}$ as well; note that (E.5.3) has the *same* coefficients as in (E.5.2). But $\mathbf{v}'_1, \dots, \mathbf{v}'_{d+1}$ are exactly the columns of the matrix that appears in Theorem E.5.5, which we have seen has linearly independent columns (because $x_1, \dots, x_d$ are pairwise different). This forces all coefficients c_j in (E.5.3) to vanish, and those are exactly the same as the coefficients in (E.5.2), so those also vanish as desired.

E.6. Change of Variables formula. As a final application of determinants, we discuss how and why determinants of derivative matrices arise as "fudge factors" when comparing integrals computed relative to different coordinate systems (e.g., standard coordinates and polar coordinates on $\mathbf{R}^2$, or standard coordinates and spherical coordinates on $\mathbf{R}^3$). This section is intended for those who have already learned about integration on regions in $\mathbf{R}^2$ and $\mathbf{R}^3$, such as from another multivariable calculus course (where the concept of a derivative matrix is usually not mentioned).

It is assumed just for this section that the reader knows about integrals of the form

$$\int_R f(x,y,z)\,dV$$

for regions R in $\mathbf{R}^3$ such as a box, a ball, a half-space (such as $z \geq 0$), or even $\mathbf{R}^3$ itself. Some books write triple integrals using the notation "$\iiint$", but that is entirely unnecessary; we stick with the shorter notation of a single integral sign, and shall often write more simply

$$\int_R f$$

when not focused on the explicit evaluation as an iteration of three single-variable integrals. Our aim is to explain how determinants of derivative matrices extend the scope of comparison formulas for such integrals computed in different coordinate systems.

We will focus on regions in $\mathbf{R}^3$ for concreteness, but the methods we discuss work for integration on regions in $\mathbf{R}^n$ for any n. The generality of $\mathbf{R}^n$ for all n is needed in probability theory when one analyzes relationships among n "random variables" for large n, as occurs in many probabilistic problems. It is also useful in some parts of physics, and in many parts of mathematics. Before we define what is meant by a *coordinate system* on a general region in $\mathbf{R}^3$, let's look at the special case of spherical coordinates:

Example E.6.1. Let B be the ball $\{\mathbf{x} \in \mathbf{R}^3 : \|\mathbf{x}\| < 7\}$ of radius 7 centered at the origin. Points $P = (x,y,z)$ of B can be described using spherical coordinates (r,θ,φ) shown in Figure E.6.1 (where the green sphere has radius $\|P\| < 7$). Here, $r < 7$ is distance from P to the origin (so $r^2 = x^2 + y^2 + z^2$), θ is angular measure in the xy-plane, and φ is angular measure from the positive z-axis. In particular, $0 \leq \theta < 2\pi$ and $0 \leq \varphi \leq \pi$. In particular, polar coordinates in the xy-plane are (r,θ).

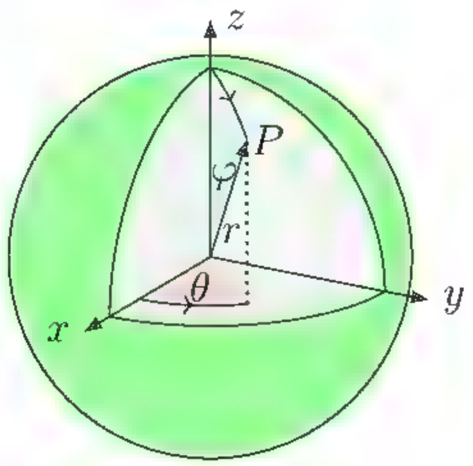


FIGURE E.6.1. Visualization of spherical coordinates for P at distance r from $\mathbf{0}$.

We are going to interpret spherical coordinates in terms of a one-to-one correspondence between points of B and points of a specific rectangular block. Consider the rectangular region

$$R = \{(r,\theta,\varphi) \in \mathbf{R}^3 : 0 < r < 7, 0 < \theta < 2\pi, 0 < \varphi < \pi\}$$

in $\mathbf{R}^3$ (a block without its faces), and define the *spherical coordinate formula* $\psi : R \to B$ by

$$\psi(r, \theta, \varphi) = (r \cos\theta \sin\varphi, r \sin\theta \sin\varphi, r \cos\varphi). \tag{E.6.1}$$

Geometric considerations with the picture in Figure E.6.1 show that if we ignore the half-planar slices $y = 0, x \geq 0$ in B then every point b that remains is described as $\psi(r, \theta, \varphi)$ for exactly one triple $(r, \theta, \varphi) \in R$.

We say ψ provides a "parameterization" of the ball B by the rectangular block R. Going in reverse from points $b \in B$ (away from where $y = 0, x \leq 0$) to points $\psi^{-1}(b) = (r(b), \theta(b), \varphi(b)) \in R$ is exactly the usual notion of spherical coordinates. ■

Now consider a general region Q in $\mathbf{R}^3$. A *parameterization* of Q amounts to specifying a second region U in $\mathbf{R}^3$ and a differentiable function $\psi : U \to Q$ so that, apart from some "thin" pieces of U and Q that have vanishing volume (such as planar slices), the function ψ has an inverse $Q \to U$ that is also differentiable. Writing $\psi(u) = (x_1(u), x_2(u), x_3(u)) \in Q$ for $u \in U$, we call ψ a *parameterization* of U much as the formula (E.6.1) describes points of a ball B by points of a rectangular block R. Writing $\psi^{-1}(q) = (y_1(q), y_2(q), y_3(q))$ for $q \in Q$, we call the functions $y_1, y_2, y_3 : Q \to \mathbf{R}$ a *coordinate system* since Example E.6.1 shows how spherical coordinates on a ball centered at $\mathbf{0}$ is a special case of this.

Our goal is to compute the integral $\int_Q f$ of a given function f over the region Q as an integral $\int_U F$ of an associated function F over the "parameter region" U. This is sometimes described as "computing $\int_Q f$ in the $\mathbf{y}$-coordinates". The crucial point is that F is *not* "f in $\mathbf{y}$-coordinates": generally $F(y_1(q), y_2(q), y_3(q)) \neq f(q)$, or in other words generally $F(u) \neq f(\psi(u))$. To formulate what to use for $F : U \to \mathbf{R}$, and why $f(\psi(u))$ doesn't work, let's return to the case of spherical coordinates.

In the study of 3-dimensional integrals in multivariable calculus using spherical coordinates, an important result is that if $f(x, y, z)$ is some function and

$$\widetilde{f}(r, \theta, \varphi) = f(r \cos\theta \sin\varphi, r \sin\theta \sin\varphi, r \cos\varphi)$$

is the same function written in spherical coordinates then

$$\int_B f = \int_0^{\pi} \int_0^{2\pi} \int_0^{7} \widetilde{f}(r, \theta, \varphi) r^2 \sin\varphi \, dr \, d\theta \, d\varphi = \int_R \widetilde{f}(r, \theta, \varphi) r^2 \sin\varphi, \tag{E.6.2}$$

where B is a ball of radius 7 centered at $\mathbf{0}$ and R is the corresponding rectangular block in terms of (r, θ, φ). In other words, the integrals of f over B and $\widetilde{f}$ over R don't agree, but instead we have to use the product of $\widetilde{f}$ by the "fudge factor" $r^2 \sin\varphi$. The need for this factor is explained by geometric arguments in multivariable calculus books, and we're going to see that the language of derivative matrices allows us to carry out a similar argument for quite general 3-dimensional coordinate systems.

First we express the fudge factor in another way. For ψ as defined by (E.6.1), direct calculation gives

$$(D\psi)(r, \theta, \varphi) = \begin{bmatrix} \cos\theta \sin\varphi & r\cos\theta\cos\varphi & -r\sin\theta\sin\varphi \\ \sin\theta\sin\varphi & r\sin\theta\cos\varphi & r\cos\theta\sin\varphi \\ \cos\varphi & -r\sin\varphi & 0 \end{bmatrix}.$$

Going through some careful algebra using that $\sin^2\theta + \cos^2\theta = 1$ and $\sin^2\varphi + \cos^2\varphi = 1$, by expanding determinants along the bottom row one finds that

$$\det(D\psi)(r, \theta, \varphi) = r^2 \sin\varphi > 0.$$

Hence, we can rewrite (E.6.2) in a form that is better-suited to going beyond spherical coordinates: since $\widetilde{f} = f \circ \psi$, it says

$$\int_B f = \int_R (f \circ \psi) |\det D\psi|$$

where the integral on each side is the conventional 3-dimensional integral for a function on a region in $\mathbf{R}^3$ (the regions being the ball B and the rectangular block R). Here is the version for a general region and any coordinate system on it:

Theorem E.6.2 (Change of Variables). Let $f : Q \to \mathbf{R}$ be a function on a region Q in $\mathbf{R}^3$, and let $\varphi : U \to Q$ be a parameterization: a differentiable function that has a differentiable inverse away from some volume-0 parts of U and Q (such as away from some planar slices). Then

$$\int_Q f = \int_U (f \circ \varphi) |\det D\varphi|.$$

If φ in coordinates is written as

$$\varphi(y_1, y_2, y_3) = (x_1(\mathbf{y}), x_2(\mathbf{y}), x_3(\mathbf{y}))$$

then the determinant factor is $\det(\partial x_i/\partial y_j)$; this is called the *Jacobian* factor. We now explain informally where the Change of Variables formula comes from, as an elegant synthesis of the approximation property of derivative matrices and of the volume interpretation of determinants discussed in Section E.3. The method we'll use carries over without any changes to the case of regions in $\mathbf{R}^n$ for any n.

First we'll approximate integrals by Riemann sums, and then use derivative matrices to approximate non-linear functions by linear ones. We ignore the volume-0 regions where φ fails to have an inverse or where some differentiability in the coordinate system or the parameterization break down. If one is careful with the approximations and the bad volume-0 slices then the informal argument below can be turned into a complete argument.

Let $B_1, B_2, \dots$ be a collection of small boxes that cover U and overlap only along their faces (which have volume 0). Then the $\varphi(B_j)$'s cover $U = \varphi(Q)$ and have volume-0 overlaps, so integrals over Q can be approximated by Riemann sums involving the $\varphi(B_j)$'s. Letting $\mathbf{b}_j \in B_j$ be a point in each box, so $\mathbf{q}_j = \varphi(\mathbf{b}_j)$ is a point in $Q_j = \varphi(B_j)$, we have

$$\int_Q f \approx \sum_j f(\mathbf{q}_j)\,\mathrm{vol}(Q_j) = \sum_j f(\varphi(\mathbf{b}_j))\,\mathrm{vol}(\varphi(B_j)) = \sum_j (f \circ \varphi)(\mathbf{b}_j)\,\mathrm{vol}(\varphi(B_j)). \tag{E.6.3}$$

Now comes the key point: how does the volume of $\varphi(B_j)$ compare to the volume of B_j? Since B_j is a small box containing the point $\mathbf{b}_j$, for points $\mathbf{x} \in B_j$ we can apply the *linear approximation property* of the derivative matrix $(D\varphi)(\mathbf{b}_j)$ at $\mathbf{b}_j$:

$$\varphi(\mathbf{x}) \approx \varphi(\mathbf{b}_j) + ((D\varphi)(\mathbf{b}_j))(\mathbf{x} - \mathbf{b}_j).$$

Letting $\mathbf{x}$ run through the points of the box B_j, we conclude that at the level of small 3-dimensional regions

$$\varphi(B_j) \approx \varphi(\mathbf{b}_j) + ((D\varphi)(\mathbf{b}_j))(B_j - \mathbf{b}_j) \tag{E.6.4}$$

with $B_j - \mathbf{b}_j$ denoting the output of subtracting $\mathbf{b}_j$ from all points of B_j.

The translation by $\varphi(\mathbf{b}_j)$ on the right side in (E.6.4) has *no effect* on volume, so

$$\mathrm{vol}(\varphi(B_j)) \approx \mathrm{vol}(((D\varphi)(\mathbf{b}_j))(B_j - \mathbf{b}_j)).$$

But for any linear transformation L and region R in $\mathbf{R}^3$, we have $\mathrm{vol}(L(R)) = |\det L|\,\mathrm{vol}(R)$! Applying this with $L = (D\varphi)(\mathbf{b}_j)$ and $R = B_j - \mathbf{b}_j$, we get

$$\mathrm{vol}(\varphi(B_j)) \approx |\det(D\varphi)(\mathbf{b}_j)|\,\mathrm{vol}(B_j - \mathbf{b}_j) = |\det(D\varphi)(\mathbf{b}_j)|\,\mathrm{vol}(B_j),$$

where the final equality uses that B_j and its translate $B_j - \mathbf{b}_j$ have the same volume. Plugging this final approximation into the right side of (E.6.3), we get

$$\int_Q f \approx \sum_j (f \circ \varphi)(\mathbf{b}_j) |\det(D\varphi)(\mathbf{b}_j)| \operatorname{vol}(B_j).$$

The right side is *exactly* a Riemann sum approximation to $\int_U (f \circ \varphi)|\det D\varphi|$ using the boxes B_j that cover U. Taking the boxes ever smaller makes the approximations even better, so passing to the limit over tinier and tinier boxes B_j turns the approximations into an equality:

$$\int_Q f = \int_U (f \circ \varphi)|\det D\varphi|,$$

as desired.

"Algebra is generous; it gives more than it is asked."

J-B. d'Alembert

F. The cross product (optional)

F.1. Introduction. This appendix discusses a special operation on 3-vectors that answers:

> **Question**: Is there a way to assign to every pair of 3-vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ a 3-vector perpendicular to each of them in a way that (i) depends *linearly* on each of $\mathbf{v}_1$ and $\mathbf{v}_2$ when the other is fixed, and (ii) is rotationally invariant (i.e., if R is a rotation of $\mathbf{R}^3$ fixing the origin, the vector assigned to $\mathbf{v}_1$ and $\mathbf{v}_2$ is carried by R to the vector assigned to $R(\mathbf{v}_1)$ and $R(\mathbf{v}_2)$)?

It turns out that there is essentially *only one* such assignment, up to an overall scaling factor; it is called the *cross product*. Everything except the statement of Theorem F.4.1 and the proofs in Section F.4 should make sense after learning the linear algebra material in Part I of this book if you take on faith some properties of determinants discussed in Appendix E (the relevant case here is $n \times n$ determinants for $n = 2, 3$). The full details build upon Appendix E and a variety of linear algebra topics up through Chapter 20.

The cross product is ubiquitous in certain physics and engineering applications (e.g., angular momentum, and the force on a charged particle due to a magnetic field), where it is traditional to use the notation

$$\mathbf{i} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{j} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{k} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

which we shall use in this appendix and call the *standard basis* of $\mathbf{R}^3$. Here is the visualization of the standard basis that is always used (by both left-handed and right-handed people).

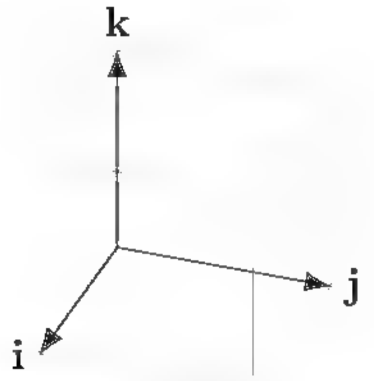


FIGURE F.1.1. The universally accepted standard basis $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$ in all scientific work

Just as it is more convenient to set up the dot product as we did in this book based on an initial algebraic definition (that was linked to a lot of geometry), it will be more convenient for us to begin with an algebraic recipe as the initial definition of the cross product. We will compute it in some examples, and then explore it in various ways via properties of 2×2 and 3×3 determinants and results about dot products and orthogonality from the main text.

Many treatments of the cross product begin with a very geometric definition in terms of a "right-hand rule". Justifying the link between the algebraic and geometric perspectives requires knowing that the algebraic perspective behaves well with respect to rotations around the origin, which is not at all apparent. One main goal of this appendix is to explain this good behavior, and another is to show that the cross product is the *unique* answer to the Question above (up to an overall scaling factor).

F.2. Definition and basic calculations.

Definition F.2.1. For 3-vectors $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$, the *cross product* $\mathbf{v} \times \mathbf{w}$ is defined to be the 3-vector

$$\mathbf{v} \times \mathbf{w} = \begin{bmatrix} v_2w_3 - v_3w_2 \\ v_3w_1 - v_1w_3 \\ v_1w_2 - v_2w_1 \end{bmatrix} = \det \begin{bmatrix} v_2 & v_3 \\ w_2 & w_3 \end{bmatrix} \mathbf{i} - \det \begin{bmatrix} v_1 & v_3 \\ w_1 & w_3 \end{bmatrix} \mathbf{j} + \det \begin{bmatrix} v_1 & v_2 \\ w_1 & w_2 \end{bmatrix} \mathbf{k}.$$

Due to the sign in front of the $\mathbf{j}$-coefficient, the expression for $\mathbf{v} \times \mathbf{w}$ in terms of $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ formally looks like the expansion along the first row of a 3×3 "determinant"

$$\mathbf{v} \times \mathbf{w} = \det \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{bmatrix} \tag{F.2.1}$$

with $\mathbf{v}$ written along the second row and $\mathbf{w}$ written along the third row. This 3×3 "determinant" shouldn't be interpreted too literally, since the entries of a determinant must be numbers and in (F.2.1) the entries in the top row are vectors. Nonetheless, (F.2.1) is a very convenient way to remember the formula for the cross product when performing calculations.

A more visual description of $\mathbf{v} \times \mathbf{w}$ called the "right-hand rule" will be given in Proposition F.4.3. As an illustration of (F.2.1), for $\mathbf{v} = 2\mathbf{i} - 4\mathbf{j} + \mathbf{k}$ and $\mathbf{w} = 3\mathbf{i} + 2\mathbf{j} + 1\mathbf{k}$ we compute

$$\begin{aligned} \mathbf{v} \times \mathbf{w} = \det \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 2 & -4 & 1 \\ 3 & 2 & 1 \end{bmatrix} &= \det \begin{bmatrix} -4 & 1 \\ 2 & 1 \end{bmatrix} \mathbf{i} - \det \begin{bmatrix} 2 & 1 \\ 3 & 1 \end{bmatrix} \mathbf{j} + \det \begin{bmatrix} 2 & -4 \\ 3 & 2 \end{bmatrix} \mathbf{k} \\ &= -6\mathbf{i} + \mathbf{j} + 18\mathbf{k}. \end{aligned}$$

A bit of experience computing cross products by hand will demonstrate how easy it is to lose minus signs and screw up such calculations.

We call $\mathbf{v} \times \mathbf{w}$ a "product" for the same reason that matrix multiplication is considered to be a "product": it is distributive over addition. This is part of the following result.

Proposition F.2.2. For any $\mathbf{v}, \mathbf{v}', \mathbf{w} \in \mathbf{R}^3$, the following hold:

(i) (distributive over vector addition) $(\mathbf{v}+\mathbf{v}')\times\mathbf{w} = \mathbf{v}\times\mathbf{w}+\mathbf{v}'\times\mathbf{w}$ and $\mathbf{w}\times(\mathbf{v}+\mathbf{v}') = \mathbf{w}\times\mathbf{v}+\mathbf{w}\times\mathbf{v}'$,
(ii) (compatibility with scalar multiplication) for any scalar c, $(c\mathbf{v}) \times \mathbf{w} = c(\mathbf{v} \times \mathbf{w}) = \mathbf{v} \times (c\mathbf{w})$,
(iii) (anti-commutative) $\mathbf{w} \times \mathbf{v} = -(\mathbf{v} \times \mathbf{w})$,
(iv) $\mathbf{v} \times \mathbf{v} = \mathbf{0}$.

The usefulness of the cross product in scientific applications and its status as the (essentially) unique answer to the Question at the start of this appendix are the primary reasons for interest in this concept. Properties (iii) and (iv) may be bothersome, but something much more distressing will arise later: the failure of associativity (see Proposition F.3.2 and the identity (F.3.4)).

PROOF. One approach is via algebraic manipulations with the definition of the cross product. We now explain an alternative perspective via properties of general determinants discussed in Appendix E.

Properties (i) and (ii) express the fact that determinants are linear in any row when the other rows are held fixed (and you can take your pick as to whether to use that fact with 2×2 determinants in the definition of $\mathbf{v} \times \mathbf{w}$ or with 3×3 determinants via (F.2.1)).

Swapping the roles of $\mathbf{v}$ and $\mathbf{w}$ in the definition of the cross product corresponds to swapping the rows in the 2×2 determinants in the definition, or alternatively in the formal 3×3 "determinant" in (F.2.1). Hence, the sign-swapping property of determinants under a change of rows explains (iii). Property (iv) expresses that determinants vanish when two rows coincide (or alternatively we can set $\mathbf{w} = \mathbf{v}$ in (iii) to get that the vector $\mathbf{v} \times \mathbf{v}$ is its own negative and so must vanish!). □

Example F.2.3. The linearity of $\mathbf{v} \times \mathbf{w}$ in each of $\mathbf{v}$ and $\mathbf{w}$ when the other is fixed is useful for computations. For example, if $\mathbf{v} = 2\mathbf{i} - 5\mathbf{j} + 3\mathbf{k}$ then

$$\mathbf{v} \times \mathbf{w} = (2\mathbf{i}) \times \mathbf{w} + (-5\mathbf{j}) \times \mathbf{w} + (3\mathbf{k}) \times \mathbf{w} = 2(\mathbf{i} \times \mathbf{w}) - 5(\mathbf{j} \times \mathbf{w}) + 3(\mathbf{k} \times \mathbf{w}).$$

The same holds with $(2, -5, 3)$ replaced by any triple of numbers. ■

This demonstrates that the most essential cross products to know how to calculate are for the standard basis against anything. But we can also run the linearity argument in the other vector variable: for any scalars w_1, w_2, w_3 we have

$$\mathbf{i} \times (w_1\mathbf{i} + w_2\mathbf{j} + w_3\mathbf{k}) = \mathbf{i} \times (w_1\mathbf{i}) + \mathbf{i} \times (w_2\mathbf{j}) + \mathbf{i} \times (w_3\mathbf{k}) = w_1(\mathbf{i} \times \mathbf{i}) + w_2(\mathbf{i} \times \mathbf{j}) + w_3(\mathbf{i} \times \mathbf{k}),$$

and similarly with "$\mathbf{i}\times$" replaced by "$\mathbf{j}\times$" or "$\mathbf{k}\times$" everywhere. In other words, what determines everything are the cross products among the standard basis vectors (and then linearity takes care of the rest). But Proposition F.2.2(ii) tells us that the self-products $\mathbf{i} \times \mathbf{i}$, $\mathbf{j} \times \mathbf{j}$, and $\mathbf{k} \times \mathbf{k}$ all vanish, so we can focus attention on cross products among distinct basis vectors. And by Proposition F.2.2(i) we only need to consider such cross products in one of the two orders; e.g., one of $\mathbf{i} \times \mathbf{j}$ or $\mathbf{j} \times \mathbf{i}$ (the other is its negative).

So finally, what determines everything are the three cross products

$$\mathbf{i} \times \mathbf{j}, \quad \mathbf{j} \times \mathbf{k}, \quad \mathbf{i} \times \mathbf{k}.$$

What are these? Flipping the third one to its negative, we have the following identities (obtained from the definition, as you should check):

$$\mathbf{i} \times \mathbf{j} = \mathbf{k}, \quad \mathbf{j} \times \mathbf{k} = \mathbf{i}, \quad \mathbf{k} \times \mathbf{i} = \mathbf{j}. \tag{F.2.2}$$

The way to remember this is to draw the symbols $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ in a circle with arrows pointing around the circle in alphabetical order like this:

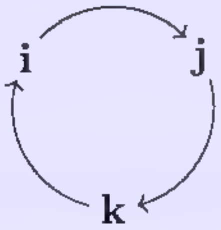


FIGURE F.2.1. The trick to remembering cross products among **i**, **j**, and **k**

Whenever we want to form a cross product $\mathbf{e} \times \mathbf{e}'$ among any two *distinct* standard basis vectors $\mathbf{e}$ and $\mathbf{e}'$, look for them next to each other in the circle. If the arrow points from $\mathbf{e}$ to $\mathbf{e}'$ then $\mathbf{e} \times \mathbf{e}'$ is the other letter apart from those two. If the arrow points the other way, take the negative of that. For instance, with $\mathbf{i} \times \mathbf{k}$ we go against the direction of the arrow linking $\mathbf{i}$ and $\mathbf{k}$, so the rule says that $\mathbf{i} \times \mathbf{k} = -\mathbf{j}$ (as is indeed the case, since $\mathbf{i} \times \mathbf{k} = -(\mathbf{k} \times \mathbf{i})$). This rule reproduces all equalities in (F.2.2).

Here is an important link between the algebra of the cross product and a geometric interpretation:

Proposition F.2.4 (Lagrange). For any $\mathbf{v}, \mathbf{w} \in \mathbf{R}^3$, we have

$$\|\mathbf{v} \times \mathbf{w}\|^2 = \|\mathbf{v}\|^2\|\mathbf{w}\|^2 - (\mathbf{v} \cdot \mathbf{w})^2, \tag{F.2.3}$$

and if $\mathbf{v}$ and $\mathbf{w}$ are nonzero with angle $\theta \in [0, \pi]$ between them then

$$\|\mathbf{v} \times \mathbf{w}\| = \|\mathbf{v}\|\|\mathbf{w}\| \sin(\theta). \tag{F.2.4}$$

For nonzero $\mathbf{v}$ and $\mathbf{w}$ and $0 < \theta < \pi$, the right side of (F.2.4) is the area of a parallelogram in $\mathbf{R}^2$ formed by vectors with lengths $\|\mathbf{v}\|$ and $\|\mathbf{w}\|$ with angle θ between them (since for the base $\mathbf{v}$ of the parallelogram the height has length $\|\mathbf{w}\| \sin(\theta)$, as one sees by drawing a picture). So in $\mathbf{R}^3$ this result says that when $\mathbf{v}$ and $\mathbf{w}$ are not scalar multiples of each other (i.e., $0 < \theta < \pi$), the magnitude of the cross product is the area of the parallelogram formed by $\mathbf{v}$ and $\mathbf{w}$ in the plane that they span.

PROOF. When $\mathbf{v}$ and $\mathbf{w}$ are nonzero, we have $\mathbf{v} \cdot \mathbf{w} = \|\mathbf{v}\|\|\mathbf{w}\| \cos(\theta)$, so (F.2.3) is asserting

$$\|\mathbf{v} \times \mathbf{w}\|^2 = \|\mathbf{v}\|^2\|\mathbf{w}\|^2 - \|\mathbf{v}\|^2\|\mathbf{w}\|^2(\cos(\theta))^2,$$

or equivalently

$$\|\mathbf{v} \times \mathbf{w}\|^2 = \|\mathbf{v}\|^2\|\mathbf{w}\|^2(1 - (\cos(\theta))^2) = \|\mathbf{v}\|^2\|\mathbf{w}\|^2(\sin(\theta))^2 = (\|\mathbf{v}\|\|\mathbf{w}\| \sin(\theta))^2.$$

Both $\|\mathbf{v} \times \mathbf{w}\|$ and $\|\mathbf{v}\|\|\mathbf{w}\| \sin(\theta)$ are non-negative (here we use that $0 \le \theta \le \pi$ to ensure $\sin(\theta) \ge 0$), so we can pass to the square roots of both sides to obtain $\|\mathbf{v} \times \mathbf{w}\| = \|\mathbf{v}\|\|\mathbf{w}\| \sin(\theta)$ as desired.

Hence, it remains to establish (F.2.3). To make the algebra as clean as possible, we will rewrite the goal in terms of dot products, which have useful algebraic properties:

$$(\mathbf{v} \times \mathbf{w}) \cdot (\mathbf{v} \times \mathbf{w}) \overset{?}{=} (\mathbf{v} \cdot \mathbf{v})(\mathbf{w} \cdot \mathbf{w}) - (\mathbf{v} \cdot \mathbf{w})(\mathbf{v} \cdot \mathbf{w}) = \det \begin{bmatrix} \mathbf{v} \cdot \mathbf{v} & \mathbf{v} \cdot \mathbf{w} \\ \mathbf{v} \cdot \mathbf{w} & \mathbf{w} \cdot \mathbf{w} \end{bmatrix}.$$

To establish this result, we are actually going to show something more general, where we allow some of the equal vectors in this equation to be more general vectors that need not be equal (opening the door to vector operations that cannot be used in the initial setup with only $\mathbf{v}$ and $\mathbf{w}$): we will show for any 3-vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{w}_1, \mathbf{w}_2$ that

$$(\mathbf{v}_1 \times \mathbf{v}_2) \cdot (\mathbf{w}_1 \times \mathbf{w}_2) \overset{?}{=} \det \begin{bmatrix} \mathbf{v}_1 \cdot \mathbf{w}_1 & \mathbf{v}_1 \cdot \mathbf{w}_2 \\ \mathbf{v}_2 \cdot \mathbf{w}_1 & \mathbf{v}_2 \cdot \mathbf{w}_2 \end{bmatrix} \tag{F.2.5}$$

If this is proved then by setting $\mathbf{w}_1 = \mathbf{v}_1$ and $\mathbf{w}_2 = \mathbf{v}_2$ we get what we want. The advantage of (F.2.5) is that it allows us to vary all four vectors, which will enable us to reduce to some special cases in a way that is not possible to do within the more constrained setting of the case $\mathbf{w}_1 = \mathbf{v}_1$ and $\mathbf{w}_2 = \mathbf{v}_2$.

One approach to proving (F.2.5) is to grind out the algebra with general entries for all four of these 3-vectors. But that is a mess, and there is a better way: much as we used the linearity of cross products to reduce all cross product calculations to those among distinct members of the standard basis, we can do something similar with (F.2.5). This is an instance of a powerful technique to establish identities in linear algebra, by exploiting linearity properties of both sides of a desired formula to reduce to a direct verification when the vectors involved come from a basis.

Before we carry this out for (F.2.5), let's explain the general context for the idea. Suppose we have two expressions $f(\mathbf{u}_1, \dots, \mathbf{u}_n)$ and $g(\mathbf{u}_1, \dots, \mathbf{u}_n)$ in n general m-vectors (these expressions f and g might be scalars or vectors in some $\mathbf{R}^N$) and we want to show they are always equal. For instance, in our setting we could use

$$f(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4) = (\mathbf{u}_1 \times \mathbf{u}_2) \cdot (\mathbf{u}_3 \times \mathbf{u}_4), \quad g(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4) = \det \begin{bmatrix} \mathbf{u}_1 \cdot \mathbf{u}_3 & \mathbf{u}_1 \cdot \mathbf{u}_4 \\ \mathbf{u}_2 \cdot \mathbf{u}_3 & \mathbf{u}_2 \cdot \mathbf{u}_4 \end{bmatrix}.$$

The key further assumption we make is that both f and g depend *linearly* on each $\mathbf{u}_i$ when the other $\mathbf{u}_j$'s are fixed. For instance, in the case $n = 4$ (as for us) this is the condition that for any $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbf{R}^m$ and varying $\mathbf{x} \in \mathbf{R}^m$, the quantities

$$f(\mathbf{x}, \mathbf{a}, \mathbf{b}, \mathbf{c}), \quad f(\mathbf{a}, \mathbf{x}, \mathbf{b}, \mathbf{c}), \quad f(\mathbf{a}, \mathbf{b}, \mathbf{x}, \mathbf{c}), \quad f(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{x})$$

all are linear as a function of $\mathbf{x}$, and similarly for g. That certainly holds in our case by inspection of the specific f and g of interest to us, due to linearity properties of the dot product and cross product in each of their vector inputs.

General linearity trick. The key observation is that in such a situation, to check that $f = g$ it is enough to do so when the vector inputs $\mathbf{u}_1, \dots, \mathbf{u}_n$ all come from within a fixed basis $\{\mathbf{e}_1, \dots, \mathbf{e}_m\}$ of $\mathbf{R}^m$. Let's explain why this works. Informally, the linearity properties of f and g in each of their inputs allows us to reconstruct their values on any $(\mathbf{u}_1, \dots, \mathbf{u}_n)$ in the *same way* from their values on such n-tuples of $\mathbf{u}_j$'s coming from the chosen basis of $\mathbf{e}_i$'s. To make this precise while keeping the notation under control, let's see it in the case $n = 4$ that we need: if we write $\mathbf{u}_1 = \sum_{i=1}^m a_i\mathbf{e}_i$ then

$$f(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4) = \sum_{i=1}^m a_i f(\mathbf{e}_i, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4), \quad g(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4) = \sum_{i=1}^m a_i g(\mathbf{e}_i, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4).$$

Thus, if we knew $f = g$ on all 4-tuples of vector inputs whose first vector is in the basis of $\mathbf{e}_i$'s, then the right sides in both of these equations would be equal term by term, hence the entire right sides would be equal to each other, and so the left sides would be equal! Hence, for the purposes of proving $f = g$, we can limit attention to 4-tuples of vectors $(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4)$ for which $\mathbf{u}_1$ is among the $\mathbf{e}_i$'s.

But now that's we've got $\mathbf{u}_1$ to be among the $\mathbf{e}_i$'s, we can run the exact same linearity argument in the second slot, to reduce to checking equality when *also* $\mathbf{u}_2$ is among the $\mathbf{e}_i$'s (in addition to $\mathbf{u}_1$ being of this type). And so on, with $\mathbf{u}_3$ and finally $\mathbf{u}_4$. That is the general linearity trick.

Coming back to our situation, it is enough to prove (F.2.5) when *each* of the four vectors belongs to the standard basis $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ of $\mathbf{R}^3$. To avoid too much case-checking, we make some more observations. First, we can assume $\mathbf{v}_1 \neq \mathbf{v}_2$ and $\mathbf{w}_1 \neq \mathbf{w}_2$ since if either equality holds then the right side of (F.2.5) is a determinant with equal rows or equal columns, while the left side vanishes since one of the cross products is a self-product and so is equal to $\mathbf{0}$. In particular, at least two members of the standard basis must appear among these four vectors, so at least one among $\mathbf{i}$ or $\mathbf{j}$ must appear. Second, swapping the roles of $\mathbf{v}_1$ and $\mathbf{v}_2$ in (F.2.5) causes each side to be multiplied by -1, as we see by inspection, and likewise for swapping the roles of $\mathbf{w}_1$ and $\mathbf{w}_2$. Finally, if you look closely at each side of (F.2.5) you'll see that if we swap both $\mathbf{v}_1$ and $\mathbf{w}_1$ as well as $\mathbf{v}_2$ and $\mathbf{w}_2$ then the value of each side is unaffected (the matrix on the right is flipped but its determinant is unaffected).

If $\mathbf{k}$ doesn't occur anywhere then the rearranging discussed above (which has the same effect on both sides and so is harmless for our needs) brings us to the case $\mathbf{v}_1 = \mathbf{i} = \mathbf{w}_1$ and $\mathbf{v}_2 = \mathbf{j} = \mathbf{w}_2$, in which case (F.2.5) says $1 \overset{?}{=} \det \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, which is true. Hence, we can assume $\mathbf{k}$ does occur, so by further rearranging as above we can arrange that either $\mathbf{v}_1 = \mathbf{i}$ or $\mathbf{v}_1 = \mathbf{j}$, and also that $\mathbf{v}_2 = \mathbf{k}$. In other words, (F.2.5) in these cases is saying that for $\mathbf{w}$ and $\mathbf{w}'$ distinct in the standard basis $\mathbf{i}, \mathbf{j}, \mathbf{k}$, we have

$$-\mathbf{j} \cdot (\mathbf{w} \times \mathbf{w}') = \det \begin{bmatrix} \mathbf{i} \cdot \mathbf{w} & \mathbf{i} \cdot \mathbf{w}' \\ \mathbf{k} \cdot \mathbf{w} & \mathbf{k} \cdot \mathbf{w}' \end{bmatrix}$$

and

$$\mathbf{i} \cdot (\mathbf{w} \times \mathbf{w}') = \det \begin{bmatrix} \mathbf{j} \cdot \mathbf{w} & \mathbf{j} \cdot \mathbf{w}' \\ \mathbf{k} \cdot \mathbf{w} & \mathbf{k} \cdot \mathbf{w}' \end{bmatrix}.$$

We'll prove these now for any $\mathbf{w}, \mathbf{w}' \in \mathbf{R}^3$. The first of these two desired equalities says that the negative of the second coefficient of $\mathbf{w} \times \mathbf{w}'$ is equal to $w_1 w_3' - w_1' w_3 = w_1 w_3' - w_3 w_1'$, which is exactly the 2×2 determinant showing up (with a sign) as the second coefficient in the *definition* of $\mathbf{w} \times \mathbf{w}'$. The second of these amounts to the fact that the expression $w_2 w_3' - w_2' w_3 = w_2 w_3' - w_3 w_2'$ is the first coefficient in the definition of $\mathbf{w} \times \mathbf{w}'$.

$\square$

Remark F.2.5. Upon learning about the cross product on $\mathbf{R}^3$, it is natural to wonder if it has an analogue on $\mathbf{R}^n$ for $n \neq 3$. The answer turns out to be essentially "no": it can be proved that an operation on $\mathbf{R}^n$ satisfying several specific properties of the cross product (among the ones discussed above) only exists for $n = 3$ and $n = 7$. There are a variety of formulations of this result, and an exposition of the proof of one such formulation that assumes no background beyond the level of this course is given in [Wal]. The cross product on $\mathbf{R}^3$ turns out to be very closely related to a special 4-dimensional "number system" called the *quaternions* that is extraordinarily useful in work with rotational dynamics in both computer graphics and aerodynamics. The exotic cross product on $\mathbf{R}^7$ is related to a much less useful and quite bizarre 8-dimensional "number system" called the *octonions*.

F.3. Triple product formulas. Having discussed some basic algebraic computations, we now turn to more sophisticated properties of the cross product. This will pave the way to its geometric significance, which is not at all evident from the initial algebraic definition (much as the initial definition of the dot product doesn't make evident at all its geometric meaning in terms of the cosine of the angle between two 3-vectors, though admittedly we were *motivated* to define the dot product in general based on deducing the formula (2.1.3) for 3-vectors as a consequence of the Law of Cosines).

The key thing which will get all subsequent considerations off the ground is the following link between cross products, dot products, and 3×3 determinants; it is called the *scalar triple product*.

Theorem F.3.1 (Scalar triple product). For any 3-vectors $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$,

$$\mathbf{u} \cdot (\mathbf{v} \times \mathbf{w}) = \det \begin{bmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{bmatrix}.$$

Moreover:

(i) $\mathbf{v} \times \mathbf{w}$ is perpendicular to both $\mathbf{v}$ and $\mathbf{w}$,
(ii) $\mathbf{v} \times \mathbf{w} = \mathbf{0}$ precisely when one of $\mathbf{v}$ or $\mathbf{w}$ is a scalar multiple of the other.

The triple product in this result is a dot product, so it is a scalar (and not a vector).

PROOF. To establish the determinantal expression for the scalar triple product, we first recall the definition of $\mathbf{v} \times \mathbf{w}$:

$$\mathbf{v} \times \mathbf{w} = \det \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{bmatrix} = \det \begin{bmatrix} v_2 & v_3 \\ w_2 & w_3 \end{bmatrix} \mathbf{i} - \det \begin{bmatrix} v_1 & v_3 \\ w_1 & w_3 \end{bmatrix} \mathbf{j} + \det \begin{bmatrix} v_1 & v_2 \\ w_1 & w_2 \end{bmatrix} \mathbf{k}.$$

Since $\mathbf{u} = u_1\mathbf{i} + u_2\mathbf{j} + u_3\mathbf{k}$, we then have

$$\mathbf{u} \cdot (\mathbf{v} \times \mathbf{w}) = u_1 \det \begin{bmatrix} v_2 & v_3 \\ w_2 & w_3 \end{bmatrix} - u_2 \det \begin{bmatrix} v_1 & v_3 \\ w_1 & w_3 \end{bmatrix} + u_3 \det \begin{bmatrix} v_1 & v_2 \\ w_1 & w_2 \end{bmatrix},$$

and this is exactly the desired determinant (expanded along its first row). This establishes the scalar triple product formula.

To prove (i), set $\mathbf{u}$ to be $\mathbf{v}$ or $\mathbf{w}$ in the scalar triple product formula. The 3×3 determinant has its first row equal to either the second or third row, so it vanishes by general properties of determinants. This establishes (i).

For (ii), we have to show two things: if one of $\mathbf{v}$ or $\mathbf{w}$ is a scalar multiple of the other then $\mathbf{v} \times \mathbf{w} = \mathbf{0}$ and otherwise $\mathbf{v} \times \mathbf{w} \neq \mathbf{0}$. When one of $\mathbf{v}$ or $\mathbf{w}$ is a scalar multiple of the other then in each 2×2 determinant appearing as a coefficient in the definition of $\mathbf{v} \times \mathbf{w}$ one of the rows is a scalar multiple of the other, so the coefficients all vanish. Hence, $\mathbf{v} \times \mathbf{w} = \mathbf{0}$ in such cases.

Now consider the case in (ii) when neither of $\mathbf{v}$ or $\mathbf{w}$ is a scalar multiple of the other. Then the pair of vectors $\{\mathbf{v}, \mathbf{w}\}$ span a plane and so extends to a basis of $\mathbf{R}^3$. Letting $\{\mathbf{v}, \mathbf{w}, \mathbf{u}\}$ be such a basis, the matrix having these as their columns has nonzero determinant (Theorem E.4.5). But determinants are unaffected by passing to the transpose matrix (turning columns into rows), and change at most by a sign when we rearrange rows and columns, so we conclude that the determinant in the scalar triple product formula for $\mathbf{u} \cdot (\mathbf{v} \times \mathbf{w})$ is nonzero, so certainly the vector $\mathbf{v} \times \mathbf{w}$ cannot be $\mathbf{0}$ (or else its dot product against this specific $\mathbf{u}$ would vanish, which we have seen is not the case). ☐

The dot product of two n-vectors is a scalar, so for $n > 1$ it does not make sense to form a "triple dot product" $\mathbf{v} \cdot \mathbf{w} \cdot \mathbf{x}$. But the cross product of two 3-vectors is again a 3-vector, so it does make sense to multiply three together, say $\mathbf{u} \times (\mathbf{v} \times \mathbf{w})$. (Don't ignore the parentheses in this expression. This will be a real issue quite soon.) We want to explore what this "triple vector product" could be. To understand how it behaves, we first consider the more typical case that neither of $\mathbf{v}$ or $\mathbf{w}$ is a scalar multiple of the other, so their span is a plane.

We have shown that $\mathbf{v} \times \mathbf{w}$ is always perpendicular to $\mathbf{v}$ and $\mathbf{w}$, and it is nonzero by Theorem F.3.1(ii), so it is a nonzero vector on the line perpendicular to the plane in $\mathbf{R}^3$ spanned by $\mathbf{v}$ and $\mathbf{w}$ and thus spans that line: $\mathrm{span}(\mathbf{v} \times \mathbf{w}) = (\mathrm{span}(\mathbf{v}, \mathbf{w}))^\perp$. But $\mathbf{u} \times (\mathbf{v} \times \mathbf{w})$ is perpendicular to $\mathbf{v} \times \mathbf{w}$ (even better, $\mathbf{u} \times \mathbf{x}$ is perpendicular to both $\mathbf{u}$ and $\mathbf{x}$ for any $\mathbf{x} \in \mathbf{R}^3$), so

$$\mathbf{u} \times (\mathbf{v} \times \mathbf{w}) \in (\mathbf{v} \times \mathbf{w})^\perp = ((\mathrm{span}(\mathbf{v}, \mathbf{w}))^\perp)^\perp = \mathrm{span}(\mathbf{v}, \mathbf{w})$$

(the final equality is the visual fact that if P is a plane through the origin in $\mathbf{R}^3$ and L is the line through $\mathbf{0}$ perpendicular to P then $L^\perp = P$). In other words, if neither of $\mathbf{v}$ or $\mathbf{w}$ is a scalar multiple of the other then

$$\mathbf{u} \times (\mathbf{v} \times \mathbf{w}) = a\mathbf{v} + b\mathbf{w} \tag{F.3.1}$$

for some scalar coefficients a and b. The same holds when one of $\mathbf{v}$ or $\mathbf{w}$ is a scalar multiple of the other: then $\mathbf{v} \times \mathbf{w} = \mathbf{0}$ and hence choosing both coefficients to be 0 works, though it will turn out that there is a "better" choice of coefficients than using 0's for all such cases.

Describing the triple vector product as in (F.3.1) brings us to a realization: $\mathbf{u} \times (\mathbf{v} \times \mathbf{w})$ is a linear combination of $\mathbf{v}$ and $\mathbf{w}$, but if we parenthesize the other way then we obtain $(\mathbf{u} \times \mathbf{v}) \times \mathbf{w} = -(\mathbf{w} \times (\mathbf{u} \times \mathbf{v}))$ that is a linear combination of $\mathbf{u}$ and $\mathbf{v}$. So typically these two "triple vector products" are different since the plane spanned by $\mathbf{v}$ and $\mathbf{w}$ is generally different from the plane spanned by $\mathbf{u}$ and $\mathbf{v}$. For example,

$$\mathbf{i} \times (\mathbf{j} \times (a\mathbf{i} + b\mathbf{j} + c\mathbf{k})) = \mathbf{i} \times (-a\mathbf{k} + c\mathbf{i}) = a\mathbf{j}$$

and

$$(\mathbf{i} \times \mathbf{j}) \times (a\mathbf{i} + b\mathbf{j} + c\mathbf{k}) = \mathbf{k} \times (a\mathbf{i} + b\mathbf{j} + c\mathbf{k}) = -b\mathbf{i} + a\mathbf{j},$$

so choosing a, b, c with $b \neq 0$ gives plenty of examples where the outcomes are different.

To summarize: the cross product is *not* associative. In contrast, though matrix multiplication is not commutative, fortunately it is associative (a fact we use all the time when working with matrix products). A lack of associativity does show up in other parts of algebra, such as exponentiation: $a^{(b^c)} \neq (a^b)^c$, and a failure to recognize this causes many algebraic errors by students when working with exponents.

The following result gives simple universal formulas for the scalar coefficients in (F.3.1), making the *failure of associativity* for the cross product quite explicit:

Proposition F.3.2. For any $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbf{R}^3$, the following hold:

$$\mathbf{u} \times (\mathbf{v} \times \mathbf{w}) = (\mathbf{u} \cdot \mathbf{w})\mathbf{v} - (\mathbf{u} \cdot \mathbf{v})\mathbf{w}$$

and

$$(\mathbf{u} \times \mathbf{v}) \times \mathbf{w} = (\mathbf{u} \cdot \mathbf{w})\mathbf{v} - (\mathbf{v} \cdot \mathbf{w})\mathbf{u}.$$

PROOF. The second formula is a reformulation of the first by rewriting $(\mathbf{u}\times\mathbf{v})\times\mathbf{w}$ as $-(\mathbf{w}\times(\mathbf{u}\times\mathbf{v}))$ and then renaming vectors suitably. So we focus on the first formula. To establish it, one approach is to write out both sides as explicit algebraic expressions in all of the vector entries and hope for the best. This works, but is tedious and not so enlightening. We will once again use the "general linearity trick" employed in the proofs of Proposition F.2.4. Both sides of the desired identity are linear in each of $\mathbf{u}, \mathbf{v}, \mathbf{w}$ when the other two are fixed, so it suffices to treat the case when each of $\mathbf{u}, \mathbf{v}, \mathbf{w}$ belong to the standard basis.

If $\mathbf{v} = \mathbf{w}$ then the two sides both vanish, so we can assume $\mathbf{v}$ and $\mathbf{w}$ are distinct members of the standard basis. Swapping the roles of $\mathbf{v}$ and $\mathbf{w}$ has the effect of multiplying both sides by -1, so it is enough to consider when $(\mathbf{v}, \mathbf{w})$ is in each of the following three cases: $(\mathbf{i}, \mathbf{j})$, $(\mathbf{j}, \mathbf{k})$, $(\mathbf{k}, \mathbf{i})$. In particular, $\mathbf{v} \times \mathbf{w}$ is the other member of the standard basis of $\mathbf{R}^3$. If $\mathbf{u}$ is distinct from both $\mathbf{v}$ and $\mathbf{w}$ (i.e., it is the other member of the standard basis of $\mathbf{R}^3$, which is $\mathbf{v} \times \mathbf{w}$ due to the three cases for $(\mathbf{v}, \mathbf{w})$ that we are now considering), then both dot products on the right side vanish and the left side also vanishes since it is the cross product of $\mathbf{v} \times \mathbf{w}$ against itself. Hence, for each $(\mathbf{v}, \mathbf{w})$ it remains to treat the cases $\mathbf{u} = \mathbf{v}$ and $\mathbf{u} = \mathbf{w}$. Thus, one of the two dot products on the right side vanishes and the other is equal to 1, so we need to prove for each of the pairs $(\mathbf{v}, \mathbf{w})$ that

$$\mathbf{v} \times (\mathbf{v} \times \mathbf{w}) = -\mathbf{w}, \quad \mathbf{w} \times (\mathbf{v} \times \mathbf{w}) = \mathbf{v}. \tag{F.3.2}$$

Since $\{\mathbf{v}, \mathbf{w}, \mathbf{v} \times \mathbf{w}\}$ is the standard basis in all of these cases (in some arrangement of those vectors), certainly these desired equalities hold up to a sign since the cross product of any two standard basis vectors of $\mathbf{R}^3$ is the other one up to a sign.

Thinking in terms of the cyclic pattern in Figure F.2.1, since $\{\mathbf{v}, \mathbf{w}, \mathbf{v} \times \mathbf{w}\}$ is a triple that goes around the circle in the direction of the arrows, the same holds for $\{\mathbf{w}, \mathbf{v} \times \mathbf{w}, \mathbf{v}\}$ and $\{\mathbf{v} \times \mathbf{w}, \mathbf{v}, \mathbf{w}\}$ (these are each a tour around the circle in the same direction as the arrows, just beginning at a different point). Hence, by that cyclic rule we have

$$\mathbf{w} \times (\mathbf{v} \times \mathbf{w}) = \mathbf{v}, \quad (\mathbf{v} \times \mathbf{w}) \times \mathbf{v} = \mathbf{w}.$$

The first of these is one of the two desired identities in (F.3.2), and the second yields the other of the desired identities in (F.3.2) upon negating both sides:

$$\mathbf{v} \times (\mathbf{v} \times \mathbf{w}) = -((\mathbf{v} \times \mathbf{w}) \times \mathbf{v}) = -\mathbf{w}.$$

$\square$

We immediately obtain the following consequence that illuminates the failure of associativity of the cross product.

Corollary F.3.3. The cross product satisfies the *Jacobi identity*: all 3-vectors $\mathbf{u}, \mathbf{v}, \mathbf{w}$ satisfy

$$(\mathbf{u} \times \mathbf{v}) \times \mathbf{w} + (\mathbf{v} \times \mathbf{w}) \times \mathbf{u} + (\mathbf{w} \times \mathbf{u}) \times \mathbf{v} = \mathbf{0}. \tag{F.3.3}$$

Equivalently, by the anti-commutativity of the cross product,

$$(\mathbf{u} \times \mathbf{v}) \times \mathbf{w} - \mathbf{u} \times (\mathbf{v} \times \mathbf{w}) = \mathbf{v} \times (\mathbf{w} \times \mathbf{u}). \tag{F.3.4}$$

If the cross product were associative then the left side of (F.3.4) would always equal $\mathbf{0}$, so the right side of (F.3.4) tells us exactly when associativity of the cross product fails: when $\mathbf{v} \times (\mathbf{w} \times \mathbf{u}) \neq \mathbf{0}$.

PROOF. We apply the second equality in Proposition F.3.2 three times:

$$\begin{aligned}
(\mathbf{u} \times \mathbf{v}) \times \mathbf{w} &= (\mathbf{u} \cdot \mathbf{w})\mathbf{v} - (\mathbf{v} \cdot \mathbf{w})\mathbf{u}, \\
(\mathbf{v} \times \mathbf{w}) \times \mathbf{u} &= (\mathbf{v} \cdot \mathbf{u})\mathbf{w} - (\mathbf{w} \cdot \mathbf{u})\mathbf{v}, \\
(\mathbf{w} \times \mathbf{u}) \times \mathbf{v} &= (\mathbf{w} \cdot \mathbf{v})\mathbf{u} - (\mathbf{u} \cdot \mathbf{v})\mathbf{w}.
\end{aligned}$$

Adding all three of these, on the right side everything cancels out (since the dot product is commutative). ☐

The way to remember the left side of (F.3.3) is that in whatever order the three vectors are put for one of the triple products, we then "cycle around" to make the other two triple products in the total sum that vanishes, just as we did in Figure F.2.1. The formulation (F.3.3) is symmetric in how it treats the three vectors, whereas (F.3.4) treats them asymmetrically (so (F.3.3) is easier to remember).

F.4. Geometric interpretation. We have established enough features of the cross product that we can now prove a crucial geometric property that is not apparent from the initial algebraic definition.

Theorem F.4.1. If A is any 3×3 matrix, then for any 3-vectors $\mathbf{v}$ and $\mathbf{w}$,

$$A^{\top}((A\mathbf{v}) \times (A\mathbf{w})) = (\det(A))(\mathbf{v} \times \mathbf{w}).$$

In particular, if A is *orientation-preserving and orthogonal* (so $A^{\top} = A^{-1}$ and $\det(A) = 1$) then

$$(A\mathbf{v}) \times (A\mathbf{w}) = A(\mathbf{v} \times \mathbf{w}).$$

PROOF. We use the same technique as in the proofs of Propositions F.2.4 and F.3.2: it is enough to check that both sides agree when $\mathbf{v}$ and $\mathbf{w}$ come from the standard basis. If $\mathbf{v} = \mathbf{w}$ then both sides vanish (since the cross product of any 3-vector against itself vanishes), so we can focus on the cases when $\mathbf{v}, \mathbf{w}$ are *distinct* members of the standard basis. The two sides behave the same way when swapping the roles of $\mathbf{v}$ and $\mathbf{w}$ (each side is multiplied by -1), so it is really enough to treat just three cases:

$$(\mathbf{v}, \mathbf{w}) = (\mathbf{i}, \mathbf{j}), (\mathbf{j}, \mathbf{k}), (\mathbf{k}, \mathbf{i}).$$

In these three cases, $\mathbf{v} \times \mathbf{w}$ is the other vector in the standard basis. Also, $A\mathbf{i}$ is equal to the first column $\mathbf{a}_1$ of A, and likewise $A\mathbf{j}$ is equal to the second column $\mathbf{a}_2$ of A and finally $A\mathbf{k}$ is equal to the third column $\mathbf{a}_3$ of A. Hence, we need to prove the three identities

$$A^{\top}(\mathbf{a}_1 \times \mathbf{a}_2) = (\det(A))\mathbf{k}, \quad A^{\top}(\mathbf{a}_2 \times \mathbf{a}_3) = (\det(A))\mathbf{i}, \quad A^{\top}(\mathbf{a}_3 \times \mathbf{a}_1) = (\det(A))\mathbf{j}$$

or more explicitly

$$A^{\top}(\mathbf{a}_1 \times \mathbf{a}_2) = \begin{bmatrix} 0 \\ 0 \\ \det(A) \end{bmatrix}, \quad A^{\top}(\mathbf{a}_2 \times \mathbf{a}_3) = \begin{bmatrix} \det(A) \\ 0 \\ 0 \end{bmatrix}, \quad A^{\top}(\mathbf{a}_3 \times \mathbf{a}_1) = \begin{bmatrix} 0 \\ \det(A) \\ 0 \end{bmatrix}. \tag{F.4.1}$$

The rows of $A^{\top}$ from top to bottom are the columns of A from left to right. In other words, in terms of rows we have

$$A^{\top} = \begin{bmatrix} \text{---} & \mathbf{a}_1 & \text{---} \\ \text{---} & \mathbf{a}_2 & \text{---} \\ \text{---} & \mathbf{a}_3 & \text{---} \end{bmatrix},$$

so via the language of matrix-vector products we have

$$A^{\top}(\mathbf{a}_r \times \mathbf{a}_s) = \begin{bmatrix} \text{---} & \mathbf{a}_1 & \text{---} \\ \text{---} & \mathbf{a}_2 & \text{---} \\ \text{---} & \mathbf{a}_3 & \text{---} \end{bmatrix} (\mathbf{a}_r \times \mathbf{a}_s) = \begin{bmatrix} \mathbf{a}_1 \cdot (\mathbf{a}_r \times \mathbf{a}_s) \\ \mathbf{a}_2 \cdot (\mathbf{a}_r \times \mathbf{a}_s) \\ \mathbf{a}_3 \cdot (\mathbf{a}_r \times \mathbf{a}_s) \end{bmatrix} \tag{F.4.2}$$

for any $1 \le r, s \le 3$.

Taking $r \neq s$, $\mathbf{a}_r \times \mathbf{a}_s$ is perpendicular to $\mathbf{a}_r$ and $\mathbf{a}_s$, so the rth and sth entries on the right side of (F.4.2) vanish. In each case in (F.4.1), this establishes the correctness of the two vanishing entries. The remaining entry in each case is a scalar triple product:

$$\mathbf{a}_3 \cdot (\mathbf{a}_1 \times \mathbf{a}_2), \quad \mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3), \quad \mathbf{a}_2 \cdot (\mathbf{a}_3 \times \mathbf{a}_1)$$

for the respective cases in (F.4.1). Thus, our task comes down to proving that each of these scalar triple products is equal to $\det(A)$. By Theorem F.3.1, these scalar triple products are the determinants of the following respective 3×3 matrices with rows as indicated:

$$\begin{bmatrix} \text{---} & \mathbf{a}_3 & \text{---} \\ \text{---} & \mathbf{a}_1 & \text{---} \\ \text{---} & \mathbf{a}_2 & \text{---} \end{bmatrix}, \quad \begin{bmatrix} \text{---} & \mathbf{a}_1 & \text{---} \\ \text{---} & \mathbf{a}_2 & \text{---} \\ \text{---} & \mathbf{a}_3 & \text{---} \end{bmatrix}, \quad \begin{bmatrix} \text{---} & \mathbf{a}_2 & \text{---} \\ \text{---} & \mathbf{a}_3 & \text{---} \\ \text{---} & \mathbf{a}_1 & \text{---} \end{bmatrix}.$$

The second of these three matrices is exactly $A^\top$, whose determinant is the same as that of A. The first is obtained from $A^\top$ via two row swaps: swap $\mathbf{a}_3$ in the third row with the second row, and then with the first row. Thus, its determinant is obtained from $\det(A^\top) = \det(A)$ via two sign changes, which is no change at all. Finally, the third matrix is the analogous double swap of rows to move the first row to the bottom, so it too has the same determinant.

The desired identity for a general 3×3 matrix A has been established, so now consider the special case when A is an orientation-preserving orthogonal matrix. Then $A^\top = A^{-1}$ and $\det(A) = 1$, so the established identity says $A^{-1}((A\mathbf{v}) \times (A\mathbf{w})) = \mathbf{v} \times \mathbf{w}$, so multiplying both sides on the left by A gives $(A\mathbf{v}) \times (A\mathbf{w}) = A(\mathbf{v} \times \mathbf{w})$, the desired result for such A. ☐

Since it is somewhat of a mouthful to keep saying "orientation-preserving rigid motion fixing the origin", for the rest of this appendix we use the following familiar terminology.

Definition F.4.2. A *rotation around the origin* is a function $R : \mathbf{R}^3 \to \mathbf{R}^3$ that fixes the origin, preserves length, and preserves orientation.

By Theorem 20.4.1 and Corollary E.4.4, these are precisely the effect of orthogonal 3×3 matrices with determinant equal to 1, and by Remark E.5.3 these always arise as rotation by some angle around a line through the origin (thereby explaining the terminology).

The refined formula $A(\mathbf{v} \times \mathbf{w}) = (A\mathbf{v}) \times (A\mathbf{w})$ in Theorem F.4.1 for rotations A around the origin is not true for general 3×3 matrices A. For example, if

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{v} = \mathbf{i}, \quad \mathbf{w} = \mathbf{j}$$

then $A(\mathbf{v} \times \mathbf{w}) = A\mathbf{k} = \mathbf{i} + \mathbf{k}$ whereas $(A\mathbf{v}) \times (A\mathbf{w}) = \mathbf{i} \times \mathbf{j} = \mathbf{k}$. In fact, Theorem F.4.1 can be used to show that for *invertible* 3×3 matrices A, if the equality $A(\mathbf{v} \times \mathbf{w}) = (A\mathbf{v}) \times (A\mathbf{w})$ holds for every $\mathbf{v}, \mathbf{w} \in \mathbf{R}^3$ then A *must* be orthogonal with determinant 1.

Finally, after all of our effort, we can establish that $\mathbf{v} \times \mathbf{w}$ (and not merely its length) is given by a purely geometric recipe when neither of $\mathbf{v}$ or $\mathbf{w}$ is a scalar multiple of the other (when one of them is a scalar multiple of the other we have $\mathbf{v} \times \mathbf{w} = \mathbf{0}$ by Proposition F.2.2(iv)).

Proposition F.4.3 (Right-hand Rule). Consider $\mathbf{v}, \mathbf{w} \in \mathbf{R}^3$ for which neither is a scalar multiple of the other (so $\mathbf{v}, \mathbf{w} \neq \mathbf{0}$). The vector $\mathbf{v} \times \mathbf{w}$ (whose magnitude is the area of the parallelogram in the plane $\mathrm{span}(\mathbf{v}, \mathbf{w})$ with corners at $\mathbf{0}, \mathbf{v}, \mathbf{w}, \mathbf{v} + \mathbf{w}$) lies on the line L through the origin perpendicular to the plane $\mathrm{span}(\mathbf{v}, \mathbf{w})$, and its direction on L is given by the "right-hand rule" defined as follows:

> Make your right hand flat, point the fingers in the direction of $\mathbf{v}$, and rotate the hand so that $\mathbf{w}$ is pointing out of your palm. Raise your right thumb so it is in the "hitchhiker" position. Then $\mathbf{v} \times \mathbf{w}$ points in the same direction along L as your right thumb.

PROOF. The right-hand rule is unaffected by rotations of space fixing the origin. Such motions are given by 3×3 orthogonal matrices with determinant equal to 1 (Theorem 20.4.1 and Corollary E.4.4), and by Theorem F.4.1 any such linear transformation A satisfies $A(\mathbf{v} \times \mathbf{w}) = (A\mathbf{v}) \times (A\mathbf{w})$. Hence, to establish the right-hand rule for computing cross products of 3-vectors $\mathbf{v}$ and $\mathbf{w}$ for which neither is a scalar multiple of the other, it suffices to check this *after* first applying any rotation around the origin to $\mathbf{v}$ and $\mathbf{w}$ together.

Here are some such motions. The plane $\mathrm{span}(\mathbf{v}, \mathbf{w})$ through the origin can be spun around the origin in space so that it becomes the xy-plane. We then rotate this plane around the origin so that $\mathbf{v}$ points along the positive x-axis. Then the vector $\mathbf{w}$ in this plane, which is not a scalar multiple of $\mathbf{v}$, has either positive or negative y-coordinate. Spin the plane in space by 180° around the x-axis if necessary so that $\mathbf{w}$ has positive y-coordinate.

We have reduced our general task to a very special case: now $\mathbf{v} = a\mathbf{i}$ with $a > 0$ and $\mathbf{w} = b\mathbf{i} + c\mathbf{j}$ with $c > 0$, so $\mathrm{span}(\mathbf{v}, \mathbf{w})$ is the xy-plane and the normal line to this through the origin is the z-axis. It remains to establish the right-hand rule in our special situation. Since $ac > 0$, so $\mathbf{v} \times \mathbf{w} = (a\mathbf{i}) \times (b\mathbf{i} + c\mathbf{j}) = (ac)\mathbf{k}$ points along the positive z-axis, we just have to check that when the right-hand rule is applied to $\mathbf{v} = a\mathbf{i}$ and $\mathbf{w} = b\mathbf{i} + c\mathbf{j}$ with $a, c > 0$ then the outcome is the positive z-axis rather than the negative z-axis. By the right-hand rule, if the fingers of the right hand point along the positive x-axis (direction of $\mathbf{v}$) and the palm is perpendicular to the xy-plane with $\mathbf{w}$ coming out of the palm at any angle strictly between 0° and 180° (not necessarily a right angle), the raised thumb points in the direction of the positive z-axis by Figure F.1.1, and we have seen that this is the direction of $\mathbf{v} \times \mathbf{w}$. $\square$

To conclude this appendix, we now confirm that the cross product is the essentially unique answer to the Question raised at the start of this appendix.

Proposition F.4.4. Consider $\mathbf{R}^3$-valued functions $f(\mathbf{v}, \mathbf{w})$ of general $\mathbf{v}, \mathbf{w} \in \mathbf{R}^3$ with the following properties:

(i) $f(\mathbf{v}, \mathbf{w})$ depends linearly on each of $\mathbf{v}$ and $\mathbf{w}$ (i.e., $f(\mathbf{v}, \mathbf{x})$ is linear in $\mathbf{x}$ for each $\mathbf{v}$, and $f(\mathbf{x}, \mathbf{w})$ is linear in $\mathbf{x}$ for each $\mathbf{w}$),

(ii) it is rotationally invariant: for every rotation R of $\mathbf{R}^3$ around $\mathbf{0}$, $f(R(\mathbf{v}), R(\mathbf{w})) = R(f(\mathbf{v}, \mathbf{w}))$.

The only such functions are scalar multiples of the cross product: there is a scalar c for which $f(\mathbf{v}, \mathbf{w}) = c(\mathbf{v} \times \mathbf{w})$ for every $\mathbf{v}, \mathbf{w} \in \mathbf{R}^3$. If we further require that the point $f(\mathbf{i}, \mathbf{j})$ on the z-axis (being a scalar multiple of $\mathbf{i} \times \mathbf{j} = \mathbf{k}$) is equal to $\mathbf{k}$ then f coincides with the cross product.

PROOF. We have seen that the cross product satisfies all of these properties. We have to show that any f at all satisfying (i) and (ii) must have the form $f(\mathbf{v}, \mathbf{w}) = c(\mathbf{v} \times \mathbf{w})$ for some scalar c. The first key step is to show that for all $\mathbf{v}, \mathbf{w} \in \mathbf{R}^3$, $f(\mathbf{v}, \mathbf{w})$ is perpendicular to $\mathbf{v}$ and $\mathbf{w}$. We separately treat the cases that $\mathbf{v}$ and $\mathbf{w}$ are linearly independent or are linearly dependent.

Suppose $\mathbf{v}$ and $\mathbf{w}$ are linearly independent. Let R be the180-degree rotation around the axis perpendicular to the plane spanned by $\mathbf{v}$ and $\mathbf{w}$, so $R\mathbf{v} = -\mathbf{v}$ and $R\mathbf{w} = -\mathbf{w}$. Then

$$R(f(\mathbf{v}, \mathbf{w})) = f(R\mathbf{v}, R\mathbf{w}) = f(-\mathbf{v}, -\mathbf{w}) = f(\mathbf{v}, \mathbf{w})$$

(the first equality uses (ii) and the final equality uses (i)), so $f(\mathbf{v}, \mathbf{w})$ lies along the axis of rotation and thus is perpendicular to $\mathbf{v}$ and $\mathbf{w}$ as desired.

If instead $\mathbf{v}$ and $\mathbf{w}$ are linear dependent then one of them is a scalar multiple of the other, and we shall show $f(\mathbf{v}, \mathbf{w}) = \mathbf{0}$. By (i) we can factor out the scalar multiplier to reduce to showing $f(\mathbf{v}, \mathbf{v}) = \mathbf{0}$ for all $\mathbf{v} \in \mathbf{R}^3$. If $\mathbf{v} = \mathbf{0}$ then the desired vanishing follows from bilinearity: $\mathbf{v} = \mathbf{0} = 0\mathbf{v}$, so $f(\mathbf{v}, \mathbf{v}) = f(0\mathbf{v}, \mathbf{v}) = 0 f(\mathbf{v}, \mathbf{v}) = \mathbf{0}$. Suppose instead $\mathbf{v} \neq \mathbf{0}$. Let R be a rotation around the axis through $\mathbf{v}$ by an angle strictly between 0 degrees and 360 degrees. Since $R\mathbf{v} = \mathbf{v}$ by how R is defined, we have

$$R(f(\mathbf{v}, \mathbf{v})) = f(R\mathbf{v}, R\mathbf{v}) = f(\mathbf{v}, \mathbf{v})$$

by (ii), so $f(\mathbf{v}, \mathbf{v})$ is fixed by R and hence lies on the axis of rotation. By design, that axis is the line spanned by $\mathbf{v}$, so $f(\mathbf{v}, \mathbf{v})$ *is a scalar multiple of* $\mathbf{v}$. We want to show this scalar multiple is $\mathbf{0}$.

Pick a line L through the origin in the plane orthogonal to $\mathbf{v}$, and let R' be a 180-degree rotation around L. We have $R'(c\mathbf{v}) = -c\mathbf{v}$ for all scalars c, since $c\mathbf{v}$ is perpendicular to L (so it lies in the plane being rotated 180 degrees by R'). Since $f(\mathbf{v}, \mathbf{v})$ is some scalar multiple of $\mathbf{v}$, it follows that $R'(f(\mathbf{v}, \mathbf{v})) = -f(\mathbf{v}, \mathbf{v})$. But (ii) gives $R'(f(\mathbf{v}, \mathbf{v})) = f(R'\mathbf{v}, R'\mathbf{v}) = f(-\mathbf{v}, -\mathbf{v}) = f(\mathbf{v}, \mathbf{v})$, the final equality due to bilinearity from (i). Thus $-f(\mathbf{v}, \mathbf{v}) = R'(f(\mathbf{v}, \mathbf{v})) = f(\mathbf{v}, \mathbf{v})$. The only vector equal to its own negative is the zero vector, so $f(\mathbf{v}, \mathbf{v}) = \mathbf{0}$ as desired. This completes the verification that $f(\mathbf{v}, \mathbf{w})$ is perpendicular to $\mathbf{v}$ and $\mathbf{w}$ for all $\mathbf{v}, \mathbf{w} \in \mathbf{R}^3$, with the additional feature that it vanishes when $\mathbf{v}$ and $\mathbf{w}$ are linearly dependent.

By the orthogonality property that we just established, $f(\mathbf{i}, \mathbf{j})$ is perpendiecular to both $\mathbf{i}$ and $\mathbf{j}$, so it belongs to the normal line to the xy-plane through the origin, which is to say the z-axis. Hence, $f(\mathbf{i}, \mathbf{j}) = c\mathbf{k}$ for some scalar c. Letting $g(\mathbf{v}, \mathbf{w}) = c(\mathbf{v} \times \mathbf{w})$ for that c, we will show $f(\mathbf{v}, \mathbf{w}) = g(\mathbf{v}, \mathbf{w})$ for all 3-vectors $\mathbf{v}$ and $\mathbf{w}$. Now we can once again invoke the general linearity trick that we have applied several times (it applies to f thanks to hypothesis (i)) to reduce to showing that

$$f(\mathbf{e}, \mathbf{e}') = c(\mathbf{e} \times \mathbf{e}') \tag{F.4.3}$$

for all choices of $\mathbf{e}$ and $\mathbf{e}'$ in the standard basis (possibly being the same basis vector).

Given a pair of perpendicular unit vectors $\mathbf{v}$ and $\mathbf{w}$ in $\mathbf{R}^3$ (such as distinct $\mathbf{e}$ and $\mathbf{e}'$ in the standard basis of $\mathbf{R}^3$), we shall build an R as in (ii) so that $R(\mathbf{v}) = \mathbf{i}$ and $R(\mathbf{w}) = \mathbf{j}$. First spin the plane $\mathrm{span}(\mathbf{v}, \mathbf{w})$ around the origin in space so it is carried onto a plane P through the origin in which the image of $\mathbf{v}$ points along the positive x-axis. Then $\mathbf{v}$ has been carried to $\mathbf{i}$ since that is the only unit vector lying on the positive x-axis. Next, we can spin the plane P around the x-axis (hence leaving the image $\mathbf{i}$ of $\mathbf{v}$ in place) so that $\mathbf{w}$ is carried to a vector along the positive y-axis. Now $\mathbf{w}$ has been carried to $\mathbf{j}$ since that is the only unit vector lying on the positive y-axis.

Letting R be the composition of these rotations, it is also such a rotation. By design, $R(\mathbf{v}) = \mathbf{i}$ and $R(\mathbf{w}) = \mathbf{j}$. Using (ii), we have

$$R(f(\mathbf{v}, \mathbf{w})) = f(R(\mathbf{v}), R(\mathbf{w})) = f(\mathbf{i}, \mathbf{j}) = c\mathbf{k} = c(\mathbf{i} \times \mathbf{j}) = c(R(\mathbf{v}) \times R(\mathbf{w})) = cR(\mathbf{v} \times \mathbf{w}) = R(c(\mathbf{v} \times \mathbf{w})),$$

where the second-to-last equality uses the invariance of the cross product under such operations R (Theorem F.4.1). Applying R^{-1} to the left and right sides of this string of equalities gives that $f(\mathbf{v}, \mathbf{w}) = c(\mathbf{v} \times \mathbf{w})$ as claimed. We have therefore proved (F.4.3) when $\mathbf{e}$ and $\mathbf{e}'$ are any different vectors in the standard basis. On the other hand, if $\mathbf{e} = \mathbf{e}'$ then we already know $f(\mathbf{e}, \mathbf{e}') = f(\mathbf{e}, \mathbf{e}) = \mathbf{0}$, and certainly $\mathbf{e} \times \mathbf{e}' = \mathbf{e} \times \mathbf{e} = \mathbf{0}$, so (F.4.3) again holds (since $\mathbf{0} = c\mathbf{0}$).

We have shown that $f(\mathbf{v}, \mathbf{w}) = c(\mathbf{v} \times \mathbf{w})$ for all $\mathbf{v}, \mathbf{w} \in \mathbf{R}^3$, with the scalar c satisfying $f(\mathbf{i}, \mathbf{j}) = c\mathbf{k}$. If we also assume $f(\mathbf{i}, \mathbf{j}) = \mathbf{k}$ then $c\mathbf{k} = \mathbf{k}$, so $c = 1$ and hence f coincides with the cross product. ☐

> “. . . research on neural networks was held up for 20 years until somebody remembered the Chain Rule!”
>
> T. Griffiths, Luce Professor of Information Technology, Consciousness, and Culture (Princeton)

> “. . . linear algebra simplifies the complex processes that underlie neural network operations, [. . .] allowing [AI models] to efficiently process vast amounts of data, recognize patterns, and make decisions.”
>
> Medium article “The Crucial Role of Mathematics in AI Development”

G. Neural networks and the multivariable Chain Rule (optional)

The ability of computers to identify high-level patterns through experience (i.e., machine-learning based on training data) has been demonstrated quite spectacularly with AlphaGo (and its descendants such as AlphaZero). This is based on the concept of a *neural network*, which we discuss here in order to identify where an essential mathematical insight occurs involving the multivariable Chain Rule.

G.1. Mathematical model of a neural network as a composition of functions. Consider the task of a computer being given a picture with 1000 pixels (encoded as a vector in $\mathbf{R}^{1000}$) and aiming to determine whether or not it is a picture of a cat. The output of the computer's work will be a single real number between 0 and 1 that measures how likely it is that the picture is of a cat.

The overall process of feeding the picture into the computer and extracting the output from the computer should be given by evaluating a function $f_{\text{cat}} : \mathbf{R}^{1000} \to \mathbf{R}$ on a 1000-vector that encodes the input of the pixel intensities. The desired features of f_{cat} are:

- if the input is a picture of a cat, f_{cat} returns a value > 0.5,
- if the input is not a picture of a cat then f_{cat} returns a value < 0.5.

This is a “classifier” problem (cat or not cat?), and the computer needs to *construct* such a function! The process of building f_{cat} inside a computer (in a way we describe below) is an instance of “machine learning”. It must be kept in mind that the function f_{cat} constructed by the computer will be an extraordinarily non-linear function, far too complicated for any human to ever discover or write down.

Two different computers working on this same “machine learning” task may build very different functions f_{cat}. All that matters is that the function does a good job at answering “cat or not cat?”; many different functions may perform equally well at this task, and we only care that the computer constructs an f_{cat} that works well in practice. The process by which f_{cat} is built inside a computer involves expressing it as a composition of many intermediate vector-valued functions (that also have to be constructed by the computer), and this all makes essential use of multivariable differential calculus, as we will explain below.

From a mathematical point of view, a *neural network* is an expression of a function $f : \mathbf{R}^n \to \mathbf{R}^m$ (often $m = 1$, as for f_{cat} above) as a composition of (typically highly non-linear) vector-valued functions:

$$\begin{tikzcd} \mathbf{R}^n \arrow[r, "\mathbf{f}_1"] & \mathbf{R}^{d_1} \arrow[r, "\mathbf{f}_2"] & \mathbf{R}^{d_2} \arrow[r, "\mathbf{f}_3"] & \ldots \arrow[r, "\mathbf{f}_{N-1}"] & \mathbf{R}^{d_{N-1}} \arrow[r, "\mathbf{f}_N"] & \mathbf{R}^m \end{tikzcd}$$

in which $\mathbf{R}^n$ on the left is called the *input layer*, $\mathbf{R}^m$ on the right is called the *output layer*, and each $\mathbf{R}^{d_i}$ in between is called a *hidden layer*. Such a composition involving N functions $\mathbf{f}_i$ is called an N*-layer* neural network (so a single-layer network involves no hidden layers and so no function composition at all).

Defining $d_0 = n$ and $d_N = m$ for convenience of notation, we refer to $\mathbf{R}^{d_j}$ as the jth *layer* (with $j = 0, 1, \ldots, N$) and call d_j the number of *nodes* in that layer. Each function $\mathbf{f}_i$ describes how to get from the output of the $(i-1)$th layer (which means just the input when $i = 1$) to the next layer. If we denote a vector in $\mathbf{R}^{d_j}$ as $\mathbf{x}_j = (x_{1,j}, \ldots, x_{d_j,j})$ then (for reasons based on analogies with the behavior of neurons in a brain) each $x_{i,j}$ is called a *neuron*, hence the name “neural network” for the overall function composition process. Many people dislike the analogies with a brain and so avoid such terminology, instead referring to each $x_{i,j}$ as a *unit* and calling the overall composition process a *multi-layer perceptron*.

The main point is that at the start f *is not known* (not even to the computer). The computer will construct (or "learn") $\mathbf{f}_i$'s so that if f is *defined* to be the composition $\mathbf{f}_N \circ \cdots \circ \mathbf{f}_1$ then it performs well at answering a specific artificial intelligence problem (e.g., cat or not cat?). Finding such $\mathbf{f}_i$'s is where multivariable calculus (e.g., gradient descent and the Chain Rule) play a critical role, as we shall see.

G.2. Building the component functions. The neural network underlying the construction of f_{cat} will define it mathematically as a composition, with the following as a toy version (in which all layers apart from the output layer have 1000 nodes):

$$\mathbf{R}^{1000} \xrightarrow{\mathbf{f}_1} \mathbf{R}^{1000} \xrightarrow{\mathbf{f}_2} \mathbf{R}^{1000} \xrightarrow{\mathbf{f}_3} \mathbf{R}^{1000} \xrightarrow{\mathbf{f}_4} \mathbf{R}. \tag{G.2.1}$$

In other words, the computer will define f_{cat} as a composition of the shape

$$\mathbf{f}_4 \circ \mathbf{f}_3 \circ \mathbf{f}_2 \circ \mathbf{f}_1 \tag{G.2.2}$$

for $\mathbf{f}_i$'s that need to be built inside the computer.

Here is a minor but potentially confusing issue. We think of input as moving from left to right when writing symbolism as in (G.2.1), but function composition notation as in (G.2.2) goes the other way around: input is first fed into the function $\mathbf{f}_1$ on the far right in (G.2.2) and we successively evaluate each function and feed its output into the next function to the left, eventually reaching the final output after evaluating the function $\mathbf{f}_4$ that is written on the far left in (G.2.2). A visualization of this is given in Figure G.2.1, motivated by consistency with the order of writing things in (G.2.2) rather than in (G.2.1) because our subsequent discussion is going to focus a lot on the mathematics of function composition.

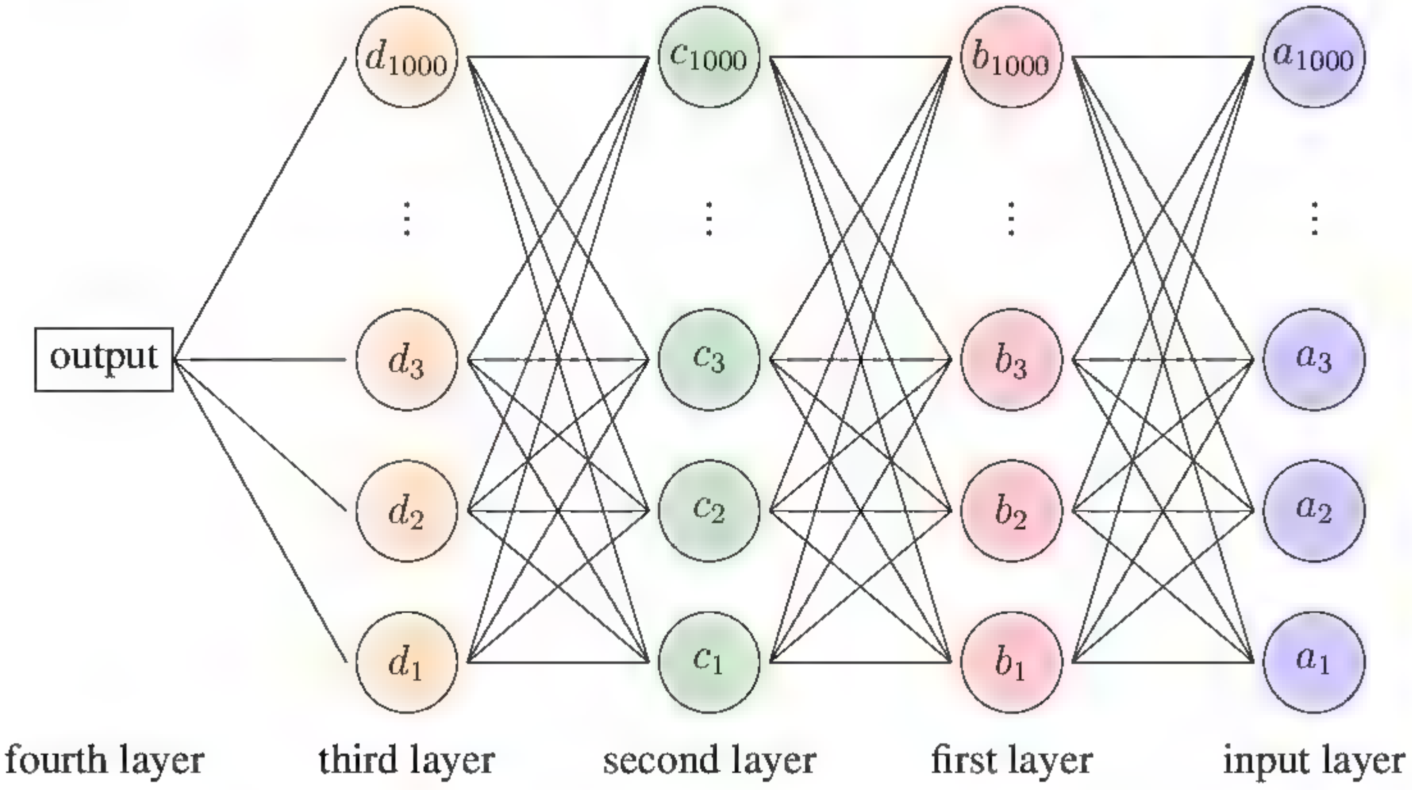


FIGURE G.2.1. A schematic of a 4-layer neural network. Circles are "neurons", lines joining neurons are "connections" among them. Going from input to output is usually drawn left to right (as in (G.2.1)), but we draw it the opposite way for consistency with (G.2.2): when we write $\mathbf{f}_4 \circ \mathbf{f}_3 \circ \mathbf{f}_2 \circ \mathbf{f}_1$, functions on the right feed into those on the left.

In Figure G.2.1, notice that there are 1000×1000 "connections" between the neurons within each pair of adjacent layers. For each $j = 1, 2, 3$ the 1000 neurons in the jth layer correspond to $x_{i,j}$ for $i = 1, 2, \ldots, 1000$ (so i keeps track of the node within a given layer), and the ith component function $f_{i,j} : \mathbf{R}^{1000} \to \mathbf{R}$ of $\mathbf{f}_j$ expresses $x_{i,j}$ as a function of the numerical values at the 1000 neurons in the previous layer. All 1000 "connections" from the neurons in the $(j-1)$th layer that feed into the ith neuron in the jth layer are regarded collectively as a single entity, expressing the functional dependence $f_{i,j}$ of $x_{i,j}$ on all 1000 values at the neurons in the previous layer.

But how does the computer determine the functions $f_{i,j}$ to be used (for a given artificial intelligence task)? For each i, $f_{i,j}$ will be taken to be a function of a special type: a composition of some linear function $w_{i,j} : \mathbf{R}^{1000} \to \mathbf{R}$ (whose coefficients are called *weights*) and a single specific non-linear function $\sigma : \mathbf{R} \to \mathbf{R}$; the same σ is used for all i (more on this below). The data of the 1000 linear "weight" functions $w_{i,j} : \mathbf{R}^{1000} \to \mathbf{R}$ of $\mathbf{x}_{j-1} \in \mathbf{R}^{1000}$ can be collected into a single 1000×1000 matrix M_j (with ith row given by the coefficients of the linear $w_{i,j}$); this encodes $\mathbf{f}_j$. (Strictly speaking, in practice $w_{i,j}$ is really a linear function plus a constant called the *bias*, but for simplicity in the present discussion we are setting every bias to be 0.) These M_j's need to be found!

Summarizing, the function $f : \mathbf{R}^{1000} \to \mathbf{R}$ used by the computer will be defined to be

$$f = \underbrace{w}_{\mathbf{f}_4:\mathbf{R}^{1000}\to\mathbf{R}} \circ \underbrace{G \circ C}_{\mathbf{f}_3} \circ \underbrace{G \circ B}_{\mathbf{f}_2} \circ \underbrace{G \circ A}_{\mathbf{f}_1} \tag{G.2.3}$$

where:

- A, B, C are 1000×1000 matrices (to be determined for the problem at hand, such as figuring out if a picture shows a cat, and in the preceding discussion are the matrices M_1, M_2, M_3),
- $w : \mathbf{R}^{1000} \to \mathbf{R}$ is a linear function (it is determined by the problem at hand, and corresponds to the 1×1000 matrix M_4 in the preceding discussion);
- $G : \mathbf{R}^{1000} \to \mathbf{R}^{1000}$ is the specific function

$$G\left(\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_{1000} \end{bmatrix}\right) = \begin{bmatrix} \sigma(y_1) \\ \sigma(y_2) \\ \dots \\ \sigma(y_{1000}) \end{bmatrix} \tag{G.2.4}$$

  where $\sigma : \mathbf{R} \to \mathbf{R}$ (called an *activation function*, due to analogies with a brain) encodes how strongly a neuron's output depends on its total input.

Historically, a popular choice of activation function was the so-called "sigmoid function" $\sigma(y) = \dfrac{1}{1+e^{-y}}$ because it nicely interpolates values from near 0 to near 1 over a range of inputs that is not too long, as shown in Figure G.2.2. In more recent years, this choice has fallen out of favor and many other options are considered in practice (such as the "rectified linear unit" function $f(x) = \max(0, x)$).

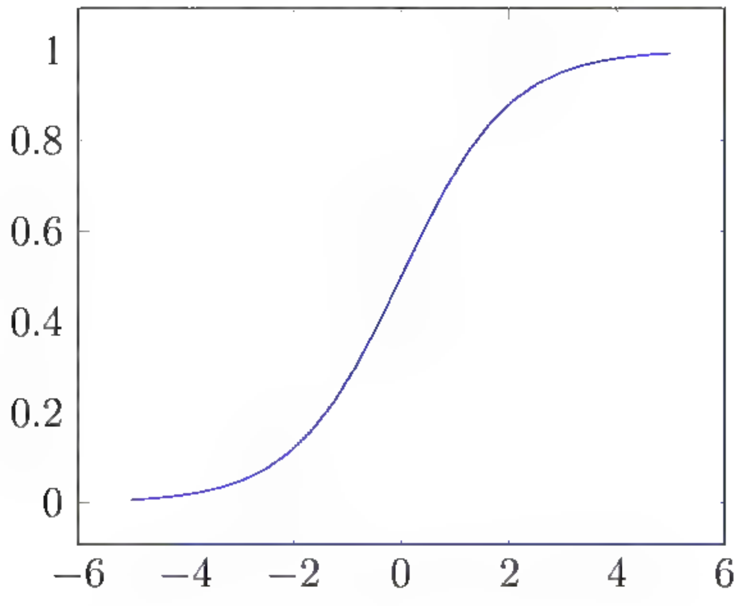


FIGURE G.2.2. The graph of $1/(1+e^{-y})$ over the interval $[-5, 5]$.

In realistic applications, one uses many more layers and the number of neurons in each layer can be on the order of millions (not 1000 as above). The output of intermediate functions (for example, the output of $G \circ B$) might be imagined as recording low-level features of the image, such as edges or color changes, but in reality the situation inside the computer is so complicated that one can't usually assign tangible meaning to specific layers.

G.3. Teaching a neural network with gradient descent. How does a neural network *learn*, by which we mean: how does it figure out good choices of A, B, C, w so that the resulting function f has the behavior we seek to be a good choice for f_{cat}? We make some initial choice for A, B, C, w to make an initial function f as above, and test this on a large body of images (called a "test set", or "training data"). For some of these images, f will correctly classify the image as cat or non-cat; on other images, it will produce the wrong answer. We then compute a "score" between 0 and 1 that measures how well the neural network did on its test:

$$E = \text{score of the neural network on the test images.} \tag{G.3.1}$$

(The testing process is automated; for the test set the correct answers are known and the performance of f on that is measured accordingly.)

We want E to be as close to 1 as possible, corresponding to getting as many images correctly classified as possible. But this is a *maximization problem in many variables* because E is really a *function* of all the parameters of the neural network: the entries of the various matrices A, B, C that appear above, as well as the coefficients of the linear function w. We will use the method of gradient descent from Section 11.3 to find A, B, C, w that collectively *minimize* the error on training data.

The gradient descent method uses a lot of partial derivatives. These will be computed by the (multivariable) Chain Rule in Theorem 17.1.5 which says in a precise way that the derivative matrix for a composition of many functions is obtained by multiplying together many derivative matrices.

G.4. The (multivariable) Chain Rule and neural networks. Our toy model for a neural network as in Figure G.2.1 defines a function of interest $f : \mathbf{R}^{1000} \to \mathbf{R}$ as a composition of the form

$$f = w \circ (G \circ C) \circ (G \circ B) \circ (G \circ A)$$

as in (G.2.3) where A, B, C are 1000×1000 matrices, G is as in (G.2.4), and $w : \mathbf{R}^{1000} \to \mathbf{R}$ is a linear function. But which A, B, C, w are best suited to the goal of the neural network? This is where "training" comes in, and multivariable optimization plays a crucial role.

Training a neural network involves *changing* entries of the matrices A, B, C and the coefficients of w in such a way that the resulting new function f_{new} after such changes performs better on test data. In other words, we run the neural network on a specific test set (evaluating f at specific inputs), and then – if the result is incorrect – we try to change A, B, C, or w to improve f. (In practice, we do this for many test inputs at once.) Thus, we need to carry out many computations to answer questions of the following sort:

> **Basic Question**: How does a small change in the upper-left entry a_{11} of A affect the output of f *when evaluated on a specific input* $\mathbf{v}_0 \in \mathbf{R}^{1000}$?

Here, note:

- The focus on a_{11} is arbitrary. In fact, we need to understand the effect on f of a small change in *any* of the matrix entries of A, B, or C, as well as in the coefficients of w.
- In practice, we use these computations to run the *gradient descent* algorithm (see Section 11.3), or a more sophisticated version of that algorithm (e.g., "stochastic gradient descent"). This takes place on $\mathbf{R}^N$ where N is *tremendously huge* (e.g., for our toy example, $N = 3 \times 1000^2 + 1000 = 3,001,000$, accounting for the matrix entries of A, B, C and the coefficients of w).

- This is a computation of a different sort than the ones usually encountered in optimization: we are not differentiating f with respect to the input! Rather, for specific numerical inputs $\mathbf{v}_0 \in \mathbf{R}^{1000}$ we are differentiating $f(\mathbf{v}_0)$ *with respect to the matrix entries of* A (viewed as parameters). To explain the idea, here is a single-variable analogue of this shift in perspective. Consider the function

  $$f_a(x) = axe^{-ax^2}$$

  for $a > 0$, where a is viewed initially as a fixed number. When we seek to make the value $f_a(2) = 2ae^{-4a}$ on the specific input $x = 2$ as big as possible, now a has become a variable! This optimization problem has the solution $a = 1/4$.

The strategy of improving artificial intelligence by adjusting the parameters of a function to make it behave better on test data has been known since long before computers were fast enough to make a practical implementation possible. Indeed, this idea was discussed by the founder of modern information theory, Claude Shannon, in a 1950 paper [**Shan1**] that he wrote about chess-playing machines (this was during the era of vacuum-tube computers; the first transistor-based computer for sale was built in 1958). In that context, Shannon wrote:

> *The chief weakness of the machine is that it will not learn by its mistakes. The only way to improve its play is by improving the program. Some thought has been given to designing a program that would develop its own improvements in strategy with increasing experience in play. Although it appears to be theoretically possible, the methods thought of so far do not seem to be very practical. One possibility is to devise a program that would change the terms and coefficients involved in the evaluation function on the basis of the results of games the machine had already played. Small variations might be introduced in these terms, and the values would be selected to give the greatest percentage of wins.*
>
> *The Gordian question, more easily raised than answered, is: Does a chess-playing machine of this type "think"? The answer depends entirely on how we define thinking.*

Remark G.4.1. One should be cautious about putting too much faith in the reliability of machine learning (see [**He**], [**Ri**]). For example, minor deviations from sample data can confuse a well-tested machine learning algorithm in ways that would never affect any human (e.g., a self-driving car can get confused by an approaching person with a red "Stop" sign on their shirt, an image classifier can "think" that a stop sign with small stickers on it is a speed limit sign, and a change in 100 random pixels in an image – imperceptible to any person – can make a computer no longer recognize an image of a cat as being a cat; also see [**Hsu**]). Debugging is also a serious problem, such as with self-driving cars. Richard Feynman remarked upon the balance between empirical and conceptual knowledge in the context of physics in his Nobel Prize lecture [**Feyn2**], and it applies equally well to machine learning today:

> *"In face of the lack of direct mathematical demonstration, one must be careful and thorough . . . , and one should make a perpetual attempt to demonstrate as much . . . as possible. Nevertheless, a very great deal more truth can become known than can be proven."*

Coming back to the Basic Question, when using $\mathbf{v}_0 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbf{R}^{1000}$ (for simplicity) it becomes this:

> How does a small change in upper-left entry a_{11} of A affect the output of f on the input $\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$?

Since $A\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}=\begin{bmatrix}a_{11}\\a_{21}\\\vdots\\a_{n1}\end{bmatrix}$, we have

$$f\left(\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}\right)=w\circ G\circ C\circ G\circ B\circ G\left(\begin{bmatrix}a_{11}\\a_{21}\\\vdots\\a_{n1}\end{bmatrix}\right).$$

We want to think of this as a function of all the matrix entries of A, B, C, and the coefficients of w, and to compute its partial derivative with respect to a_{11}. The Chain Rule gives a product of *lots* of matrices and a column vector:

$$(\text{partial derivative with respect to } a_{11})=\underbrace{(Dw)}_{1\times 1000}\times\underbrace{(DG)}_{1000\times 1000}\times\underbrace{(DC)}_{1000\times 1000}\times(DG)\times(DB)\times(DG)\times\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}.$$

WARNING! We are being a bit sloppy here with the notation. The Chain Rule really gives us specific points at which to evaluate each of the derivative matrices of w, G, C and so on, on the right side. To keep the notation from exploding out of control, we have not written the evaluation points. Thus, for example, *there are actually three different matrices for the three occurrences of DG in the above equation.* An important aspect is that each of these three matrices denoted as DG (evaluated at various points) is *diagonal* since the ith entry of $G(\mathbf{x})\in\mathbf{R}^{1000}$ is the function $G_i(\mathbf{x})=\sigma(x_i)$ that depends only on x_i (so for $j\neq i$ the off-diagonal entry $\partial G_i/\partial x_j$ vanishes). In some treatments of neural networks, you will not find this diagonal matrix written out explicitly; instead, an extra notion called the "Hadamard product" is introduced as a shorthand for the multiplication of a vector by a diagonal matrix (also see Remark 14.3.10).

What if we had asked exactly the same question, but with a_{11} replaced by a_{12}? The result comes out similarly except that the column vector on the right changes to have its entry "1" appear in the second position rather than the first:

$$(\text{partial derivative with respect to } a_{12})=\underbrace{(Dw)}_{1\times 1000}\times\underbrace{(DG)}_{1000\times 1000}\times\underbrace{(DC)}_{1000\times 1000}\times(DG)\times(DB)\times(DG)\times\begin{bmatrix}0\\1\\0\\\vdots\\0\end{bmatrix}.$$

There are similar formulas for other partial derivatives.

Our end conclusion is like this: if we can compute the matrix product

$$\underbrace{(Dw)}_{1\times 1000}\times\underbrace{(DG)}_{1000\times 1000}\times\underbrace{(DC)}_{1000\times 1000}\times(DG)\times(DB)\times(DG) \tag{G.4.1}$$

then we can easily compute the partial derivative of the output with respect to any matrix entry of A. There is a similar formula for partial derivatives with respect to matrix entries of B or C or the coefficients of w; they involve fewer matrix multiplications (i.e., we drop the last few terms in (G.4.1)), for much the same reason that when computing a partial derivative of the form $\frac{\partial}{\partial x}(f(g(x,h(y,k(z)))))$ via the Chain Rule everything involving the functions $h(y_1,y_2)$ and $k(z)$ is carried along like a constant; i.e., no derivatives of h or k intervene.

Note that the Chain Rule enables us to compute derivatives of a composite function **without** grinding out a huge explicit formula for the function. This is useful in single-variable calculus (e.g., we can differentiate $(x+3)^{20}$ as $20(x+3)^{19}$ without first needing to explicitly expand $(x+3)^{20}$ via the Binomial Theorem). In a neural network setting with more than a few layers, the Chain Rule is even more essential for computing the partial derivatives that arise. But there is something else about the Chain Rule that is important for time efficiency, as we now discuss.

G.5. Left-handed or right-handed multiplication? In practice, the derivatives are computed by an algorithm called *backpropagation* (which we are not going to describe in detail, but rests on the Chain Rule via expressions as in (G.4.1)), and we want to point out a mathematical observation about multiplying matrices that conveys a sense of what makes backpropagation very efficient in practice.

Suppose you are given the matrix product

$$\begin{bmatrix} 3 & -1 & 5 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 1 & 4 \\ 3 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 & 1 \\ 3 & -1 & 0 \\ -2 & 0 & -1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -1 \\ 1 & 1 & 3 \\ 0 & -1 & 4 \end{bmatrix}$$

Note that this looks a lot like (G.4.1): it has a row vector followed by several matrices. In (G.4.1) the vectors and matrices are much bigger (1000 entries in each row and column rather than 3) but the same basic principles are going to apply. **Please evaluate this product yourself before reading on; this will help you better appreciate the discussion that follows.**

How did you do it? There are two natural ways to evaluate the matrix product: left to right, and right to left. Proceeding left to right we get

$$\begin{bmatrix} 16 & 4 & -2 \end{bmatrix} \begin{bmatrix} 0 & 2 & 1 \\ 3 & -1 & 0 \\ -2 & 0 & -1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -1 \\ 1 & 1 & 3 \\ 0 & -1 & 4 \end{bmatrix} = \begin{bmatrix} 16 & 28 & 18 \end{bmatrix} \begin{bmatrix} 2 & 1 & -1 \\ 1 & 1 & 3 \\ 0 & -1 & 4 \end{bmatrix} = \begin{bmatrix} 60 & 26 & 140 \end{bmatrix}.$$

Proceeding right to left we get

$$\begin{bmatrix} 3 & -1 & 5 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 1 & 4 \\ 3 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 10 \\ 5 & 2 & -6 \\ -4 & -1 & -2 \end{bmatrix} = \begin{bmatrix} 3 & -1 & 5 \end{bmatrix} \begin{bmatrix} 6 & 2 & 12 \\ -7 & 0 & 6 \\ 7 & 4 & 22 \end{bmatrix} = \begin{bmatrix} 60 & 26 & 140 \end{bmatrix}.$$

Viewed on paper (or on a computer screen), these look almost the same. However, if you did them by hand, you will notice that the *second way is much more time-consuming than the first*:

- done the first way (left to right), we perform three matrix-vector multiplications.
- done the second way (right to left), we perform two matrix multiplications and one matrix-vector multiplication.

Clearly, multiplying a matrix by another matrix involves many more computations than multiplying a matrix by a vector (for $n \times n$ matrices and vectors in $\mathbf{R}^n$, the matrix-matrix product involves n^3 multiplications whereas the matrix-vector product involves n^2 multiplications). Therefore, left-to-right is faster than right-to-left, and if the number n of entries in the vectors is on the order of millions then the extra efficiency is significant in practice.

The mathematical idea of backpropagation is to evaluate (G.4.1), and all the related expressions that occur when computing partial derivatives of the output of a neural network, by left-to-right multiplication rather than right-to-left. Going from left to right might seem like the most natural thing in the world when the computation is presented as a pure linear algebra multiplication problem as above, but the context in which this arises for neural networks doesn't make this choice so natural-looking at all! Recall that the actual situation of interest arises from applying the Chain Rule to a function composition such as $\mathbf{f}_4(\mathbf{f}_3(\mathbf{f}_2(\mathbf{f}_1(\mathbf{x}))))$, and when evaluating such a composite function we have to begin on the *right* with $\mathbf{f}_1$ and work our way to the left by successive evaluation.

But for the computation of a product of derivative matrices (as emerge from the Chain Rule), the **associativity** of matrix multiplication allows us to alternatively compute things *beginning on the left* and working our way to the right. When interpreted in terms of the neural network, this is often referred to as "propagating errors backward from output to input" (hence the name "backpropagation").

This combination of the associativity of matrix multiplication and the Chain Rule expressed in terms of matrix multiplication is also relevant to the design of highly efficient neural networks beyond scalar-valued output (as with $f_{\text{cat}} : \mathbf{R}^{1000} \to \mathbf{R}$). We'll explain this in the context of auto-encoders used for "dimension reduction" (in facial recognition and image compression, for example), which can involve a neural network seeking to "learn" a vector-valued $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^n$ via an d-layer neural network of the form

$$\mathbf{R}^n \xrightarrow{\mathbf{f}_1} \mathbf{R}^n \xrightarrow{\mathbf{f}_2} \mathbf{R}^n \xrightarrow{\mathbf{f}_3} \dots \xrightarrow{\mathbf{f}_{d-1}} \mathbf{R}^n \xrightarrow{\mathbf{f}_d} \mathbf{R}^n,$$

so for the optimization work it is necessary to compute the $n \times n$ derivative matrices

$$D_k = D(\mathbf{f}_d \circ \cdots \circ \mathbf{f}_k)$$

(evaluating at specific points) for $1 \leq k \leq d$. The Chain Rule tells us

$$D_k = M_d(M_{d\ \ 1}(M_{d\ \ 2}(\cdots(M_{k+1}M_k)\cdots))) \tag{G.5.1}$$

where $M_j = D\mathbf{f}_j$ (evaluated at an appropriate point); the parentheses here match the order in which one would compute the corresponding function composition.

Here is the key insight. Since (G.5.1) involves $d - k$ matrix products, the determination of all the D_k's requires $(d-1) + (d-2) + \cdots + 1 = d(d-1)/2 \approx (1/2)d^2$ matrix products when computed via the parenthesization as in (G.5.1). But the **associativity** of matrix multiplication allows us to massively cut down on the number of such products to compute: we can rewrite the parentheses in (G.5.1) as

$$D_k = (((\cdots(M_d M_{d\ \ 1})M_{d\ \ 2})\dots)M_{k+1})M_k = D_{k+1}M_k.$$

Aha! So instead of $\approx (1/2)d^2$ products of $n \times n$ matrices to compute the D_k's, we only need to compute $\approx d$ of them:

$$D_d = M_d, D_{d\ \ 1} = D_d M_{d\ \ 1}, D_{d\ \ 2} = D_{d\ \ 1}M_{d\ \ 2}, \dots, D_2 = D_3M_2, D_1 = D_2M_1.$$

When combined with modern algorithms for efficient matrix multiplication (tailored to matrices with many 0's in the case of convolution neural networks), this provides the tremendous time savings that makes backpropagation work efficiently in practice.

> “An algorithm must be seen to be believed, and the best way to learn what an algorithm is all about is to try it.”
>
> D. Knuth [Kn, Sec. 1.1, p. 4]

H. The QR algorithm (optional)

As is mentioned in Remark 23.3.7, a fundamental application of the QR-decomposition from Chapter 22 is the QR algorithm: this is how computers accurately compute the eigenvalues of $n \times n$ matrices, and it is regarded as one of the 10 most important numerical algorithms developed in the 20th century.

This algorithm was invented independently and simultaneously during the time period 1958-1962 by Vera Kublanovskaya[32] in the Soviet Union and by John Francis[33] in England. Curiously, the early applications of this work in the Soviet Union were limited to $n \leq 5$ with calculations done by hand or with calculators; it was not put on electronic computers or used for applications outside mathematics itself [GU, Sec. 3.4]. It must be emphasized that the algorithm does not assume the matrix is symmetric; this is important for applications. Credit for refining the QR algorithm into a powerful and widely applicable tool goes largely to J.H. Wilkinson for his book [Wil1] (within one year of its publication in 1965, a book review [Pa] of this work said: “Long awaited and much needed, this book is already hard to procure.”).

We shall describe the most basic version of the QR algorithm and why it works, as an application of orthogonality in linear algebra. The algorithm must be refined in several ways in order to work efficiently and be numerically stable (i.e., well-behaved under slight changes in the input data, to allow for experimental error in measurements). The issue of numerical stability is absolutely critical: many mechanical engineering designs (such as for airplanes) involve solving eigenvalue problems, so one needs confidence in the numerics before the design begins. (One impetus for the creation of the algorithm was for applications in aircraft design [GU, p. 469].)

H.1. The algorithm. Assume A is invertible with n eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ in $\mathbf{R}$ that have different absolute values, arranged so that $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$. Imposing such a hypothesis on the eigenvalues for an algorithm aimed at accurately computing the eigenvalues may seem circular, but it is not because there are ways to check it without knowing the eigenvalues. More importantly, these hypotheses on A can be completely bypassed by modifying the algorithm (via a technique called “Wilkinson shifts”), and this is done for all implementations of the algorithm.

Define $A_0 = A$ and write its QR-decomposition as $A_0 = Q_0 R_0$. Now flip the order of multiplication! That is, define $A_1 = R_0 Q_0$.

The matrices A_1 and A_0 typically have little to do with each other. But they have one thing in common: the same eigenvalues! In fact, the products BC and CB of any two $n \times n$ matrices B and C have the same eigenvalues. For invertible B we have $BC = (BC)BB^{-1} = B(CB)B^{-1}$, and the eigenvalues of any $n \times n$ matrix M match those of BMB^{-1} (because multiplication by the invertible B carries eigenvectors of M exactly over to those of BMB^{-1} while preserving the eigenvalue: $M\mathbf{v} = \lambda\mathbf{v}$ precisely when $BM\mathbf{v} = B(\lambda\mathbf{v}) = \lambda(B\mathbf{v})$, yet $BM\mathbf{v} = (BMB^{-1})(B\mathbf{v})$). Limit arguments allow to pass from invertible B to general B (for algebraic reasons that we omit).

---

[32] Vera Kublanovskaya (1920-2012) worked in numerical linear algebra and other areas of applied mathematics. In additional to her work on the QR algorithm, she studied inverse eigenvalue problems and developed a method of computing the Jordan canonical form of a matrix. She was mathematically active into her 90’s and finished her last paper on the day before she passed away.

[33] John Francis (b. 1934) is a retired British computer scientist who left the field of numerical analysis almost immediately after inventing the QR algorithm in the early 1960’s. He worked for a variety of computer companies during his career, and was entirely unaware of the tremendous practical importance of the QR algorithm until nearly 50 years after its invention.

Next, form the QR-decomposition of A_1, written as $A_1 = Q_1R_1$. Now flip again! That is, define $A_2 = R_1Q_1$ (so its eigenvalues match those of A_1, which in turn match those of $A_0 = A$), and form the QR-decomposition of this, written as $A_2 = Q_2R_2$. Rinse and repeat, getting successive $n \times n$ matrices $A_m = R_{m-1}Q_{m-1}$ whose QR-decomposition we write as Q_mR_m and then use those to define $A_{m+1} = R_mQ_m$, and so on. All A_m's have the same eigenvalues as A.

The pleasant surprise is that, subject to a very mild further assumption (which we'll get to below, and becomes unnecessary under a modification of the algorithm by incorporating "Wilkinson shifts", as mentioned above), the matrices A_m "converge" entrywise to an upper triangular matrix. But the common eigenvalues for A and the A_m's are inherited by this limit matrix, and for an upper triangular matrix it can be shown that the eigenvalues are exactly the diagonal entries. Hence, the diagonal entries of this limit matrix are the eigenvalues we sought!

Remark H.1.1. The convergence to an upper triangular matrix is where one needs the initial assumption concerning eigenvalues in $\mathbf{R}$ with different absolute values. The convergence can fail otherwise; e.g., for $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ with eigenvalues 1 and -1 one has $A_m = A$ for all m, so in this case the above procedure accomplishes nothing. There are ways around this, by modifying the algorithm.

H.2. Why does it work? What is the real meaning of the maneuver of flipping the order of multiplication of Q and R at each step, and why does this strange-looking process converge to an upper triangular matrix? We now sketch an answer to this, and refer to [Wat] for further details.

Let $\mathbf{v}_i$ be an eigenvector of A for the eigenvalue λ_i (recall that $|\lambda_1| > |\lambda_2| > \dots$), and define $V_k = \mathrm{span}(\mathbf{v}_1, \dots, \mathbf{v}_k)$, so V_k is k-dimensional and $V_1 \subset V_2 \subset \cdots \subset V_n = \mathbf{R}^n$. These V_k's satisfy the following remarkable property, sometimes referred to as "power iteration". It illustrates the phenomenon that big powers of a matrix are controlled by the eigenvalues and eigenvectors.

Proposition H.2.1. Let W be *any* nonzero linear subspace of $\mathbf{R}^n$, with some dimension k. Assume W satisfies the "general position" condition that it doesn't contain any nonzero vectors in the subspace $\mathrm{span}(\mathbf{v}_{k+1}, \dots, \mathbf{v}_n)$ with complementary dimension $n - k$ (this holds for "most" k-dimensional W). As we repeatedly apply A to the entirety of W over and over again (i.e., for each $m \geq 1$ we consider the linear subspace $A^m(W)$ consisting of the vectors of the form $A^m(\mathbf{w})$ for $\mathbf{w} \in W$), the k-dimensional subspaces $A^m(W)$ in $\mathbf{R}^n$ for $m = 1, 2, \dots$ get arbitrarily "close" to V_k.

The notion of "closeness" for linear subspaces of $\mathbf{R}^n$ with a common dimension k as in this proposition means that every unit vector in one of them is very close to a unit vector in the other. In the case of lines through the origin, which is to say $k = 1$, this means that the two lines have a very small angle between them. For planes in $\mathbf{R}^3$ through the origin, this notion of closeness is the same as the normal lines to the planes having a very small angle between them.

PROOF. The validity of the proposition in for general k can be reduced to the case $k = 1$; this may sound very surprising, but it can be shown using a trick that we omit (involving determinants), so we focus on the case $k = 1$. Now $W = \mathrm{span}(\mathbf{w})$ is a line, and we can write

$$\mathbf{w} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_n\mathbf{v}_n$$

for some scalars $c_1, \dots, c_n$. We have $c_1 \neq 0$ by the "general position" hypothesis on W, and $A^m(W)$ is the span of $A^m(\mathbf{w})$.

We compute

$$A^m(\mathbf{w}) = c_1A^m(\mathbf{v}_1) + c_2A^m(\mathbf{v}_2) + \cdots + c_nA^m(\mathbf{v}_n) = c_1\lambda_1^m\mathbf{v}_1 + c_2\lambda_2^m\mathbf{v}_2 + \cdots + c_n\lambda_n^m\mathbf{v}_n.$$

The line this spans is unaffected by dividing it by λ_1^m. But when we do that division we get

$$c_1\mathbf{v}_1 + c_2(\lambda_2/\lambda_1)^m\mathbf{v}_2 + \cdots + c_n(\lambda_n/\lambda_1)^m\mathbf{v}_n$$

with the coefficients $(\lambda_j/\lambda_1)^m$ for $j = 2, \dots, n$ all *very close to* 0 for large m since $|\lambda_j/\lambda_1| = |\lambda_j|/|\lambda_1| < 1$. (For any number with absolute value < 1, its high powers rapidly approach 0.)

Hence, the line $A^m(W) = \mathrm{span}(A^m(\mathbf{w}))$ is very close to the line $\mathrm{span}(c_1\mathbf{v}_1) = \mathrm{span}(\mathbf{v}_1) = V_1$, as desired. (This final step uses the "general position" condition $c_1 \neq 0$.) $\square$

To exploit this proposition, we take W to be $W_k = \mathrm{span}(\mathbf{e}_1, \dots, \mathbf{e}_k)$. Applying the proposition to W_k requires a "general position" condition on the collection of $\mathbf{e}_i$'s relative to the eigenvectors $\mathbf{v}_j$, which we now assume (in practice it can be avoided by modifying the algorithm).

Apply Gram–Schmidt to the basis $A^m(\mathbf{e}_1), \dots, A^m(\mathbf{e}_k)$ of $A^m(W_k)$ to get an orthogonal basis of $A^m(W_k)$. The vectors in this orthogonal basis will usually be very long or very short for large m (depending on the $|\lambda_i|$'s), so we get rid of that effect by dividing the output vectors by their lengths to get an orthonormal basis $\{\mathbf{q}_{1,m}, \dots, \mathbf{q}_{k,m}\}$ of $A^m(W_k)$. These unit vectors must get very close to unit vectors in V_k since $A^m(W_k)$ is very close to V_k by Proposition H.2.1.

Since the first r vectors emerging from Gram–Schmidt only ever depend on the first r vectors of the input, when we run the same procedure for $A^m(W_{k+1})$ the first k vectors in the resulting orthonormal basis are *the same* as the k vectors $\mathbf{q}_{1,m}, \dots, \mathbf{q}_{k,m}$ obtained for $A^m(W_k)$. In other words, these $\mathbf{q}_{i,m}$'s built for different k all fit together coherently, as part of an overall orthonormal basis $\mathbf{q}_{1,m}, \dots, \mathbf{q}_{n,m}$ of $A^m(W_n) = \mathbf{R}^n$.

The wonderful fact is that, up to a sign, we can pin down *exactly which unit vector* in V_k each unit vector $\mathbf{q}_{i,m}$ is very close to for each $1 \leq i \leq k$:

Proposition H.2.2. Let $\{\mathbf{w}_1, \dots, \mathbf{w}_n\}$ be the output of Gram–Schmidt applied to $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$, so the unit vectors $\mathbf{w}'_i = \mathbf{w}_i/\|\mathbf{w}_i\|$ constitute an orthonormal basis of $\mathbf{R}^n$ whose first k members span V_k.

Each $\mathbf{q}_{i,m}$ gets very close to one of $\pm\mathbf{w}'_i$ (sign maybe depending on m).

PROOF. Since the first r vectors emerging from Gram–Schmidt only depend on the first r vectors of the input, to analyze anything about $\mathbf{q}_{i,m}$ only $A^m(\mathbf{e}_1), \dots, A^m(\mathbf{e}_i)$ are relevant. This says that we only need to focus on $A^m(W_i)$. In other words, we can rename i as k and focus on the case $i = k$, assuming (when $k > 1$) that the proposition is established for all smaller values of i.

First consider $k = 1$. Then $\pm\mathbf{w}'_1$ are the *only* unit vectors in V_1 since a line through the origin only has two unit vectors. But $A^m(W_1)$ is very close V_1 (for big m) by Proposition H.2.1, so the unit vector $\mathbf{q}_{1,m}$ in $A^m(W_1)$ must be close to one of the only two unit vectors $\pm\mathbf{w}'_1$ in V_1. This settles the case $k = 1$, and so also settles the case $i = 1$ in general.

Next, suppose $k = 2$. The unit vector $\mathbf{q}_{2,m}$ is orthogonal to $\mathbf{q}_{1,m} \approx \pm\mathbf{w}'_1$, so whatever unit vector in V_2 it is very close to must be essentially orthogonal to $\mathbf{w}'_1$ and so very close to a unit vector in the orthogonal complement of $\mathrm{span}(\mathbf{w}'_1)$ in V_2. Here we are repeatedly using Theorem 6.2.4 and the Pythagorean Theorem in $\mathbf{R}^n$ (Theorem 2.3.1), and hopefully the geometric intuition is plausible just by drawing pictures. The orthogonal complement in the plane V_2 to the line $\mathrm{span}(\mathbf{w}'_1) = V_1$ is a *line*, so $\pm\mathbf{w}'_2$ are the only two unit vectors in that orthogonal complement. Hence, $\mathbf{q}_{2,m} \approx \pm\mathbf{w}'_2$ for big m.

And on it goes similarly for larger k, basically because the orthogonal complement to V_{k-1} in V_k is always a line (since $\dim V_k = 1 + \dim V_{k-1}$), so that orthogonal complement must be spanned by the unit vector $\mathbf{w}'_k$ that it contains. Hence, this orthogonal complement has $\pm\mathbf{w}'_k$ as its only unit vectors, so $\mathbf{q}_{k,m}$ must be close to one of them when m is big. $\square$

By design of the QR-decomposition in terms of the Gram–Schmidt process, if we write the QR-decomposition of A^m as $Q'_m R'_m$ then the columns of Q'_m are exactly $\mathbf{q}_{1,m}, \dots, \mathbf{q}_{n,m}$ from left to right. But at the cost of some signs, Proposition H.2.2 tells us that these vectors are converging entrywise to the vectors $\mathbf{w}'_1, \dots, \mathbf{w}'_n$ *that have nothing to do with* m, so at the cost of some signs the orthogonal matrices Q'_m for big m are all getting *very close to each other* (in the sense of each matrix entry).

To leverage this knowledge about Q'_m for big m, we need some other way to describe the QR-decomposition of A^m. It is *precisely* at this step that the weird maneuver of flipped products in the QR algorithm emerges. To see what is going on, let's consider $m = 4$: we have

$$A^4 = AAAA = Q_0R_0Q_0R_0Q_0R_0Q_0R_0 = Q_0(R_0Q_0)(R_0Q_0)(R_0Q_0)R_0,$$

using associativity of matrix multiplication at this final step. But by design, the flipped product R_0Q_0 is what we call A_1, which in turn has QR-decomposition denoted Q_1R_1, so substituting this and using the same associativity trick gives

$$A^4 = Q_0(Q_1R_1)(Q_1R_1)(Q_1R_1)R_0 = Q_0Q_1(R_1Q_1)(R_1Q_1)R_1R_0.$$

But now again we play the same game: R_1Q_1 is what we call A_2, whose QR-decomposition is denoted Q_2R_2, and so we substitute this to get

$$A^4 = Q_0Q_1(Q_2R_2)(Q_2R_2)R_1R_0 = Q_0Q_1Q_2(R_2Q_2)R_2R_1R_0.$$

One last pass: R_2Q_2 is what we call A_3, whose QR-decomposition is denoted Q_3R_3, so substituting this and using associativity gives

$$A^4 = Q_0Q_1Q_2(Q_3R_3)R_2R_1R_0 = (Q_0Q_1Q_2Q_3)(R_3R_2R_1R_0).$$

Any product of orthogonal matrices is orthogonal, and any product of upper triangular matrices with positive diagonal entries is again such a matrix, so $Q_0Q_1Q_2Q_3$ is orthogonal and $R_3R_2R_1R_0$ is upper triangular with positive diagonal entries. But the QR-decomposition of an invertible matrix is uniquely determined (this essentially expresses that Gram–Schmidt is an unambiguous algorithm), so the factors Q'_4, R'_4 of the QR-decomposition of A^4 must agree with these 4-fold products: $Q'_4 = Q_0Q_1Q_2Q_3$, $R'_4 = R_3R_2R_1R_0$. This works the same way with "4" replaced by any m, so

$$Q'_m = Q_0Q_1 \cdots Q_{m-1}.$$

But we have already seen that up to some signs, the matrices Q'_m all get very close to each other for large m. Since $Q'_{m+1} = Q'_m Q_m$, it follows from the closeness of Q'_{m+1} and Q'_m up to some signs that if m is big then Q_m is *close to the identity matrix* up to some signs.

By definition Q_mR_m is the QR-decomposition of A_m, so up to some signs we conclude that $A_m \approx R_m$ for large m. Changing the entries of an upper triangular matrix by some signs doesn't affect it being upper triangular (though it may affect positivity conditions on diagonal entries), so A_m is very close to an upper triangular matrix for big m! Since the eigenvalues of an upper triangular matrix are exactly its diagonal entries, it can be deduced (using characteristic polynomials as in Section E.5, so that theoretical device is lurking in the justification of the QR algorithm) from the nearness of A_m to an upper triangular matrix for big m that the diagonal entries of the A_m's are getting very close to the eigenvalues of A_m for big m. But those in turn are exactly the eigenvalues of A, so we conclude that for big m the diagonal entries of A_m are very close to the eigenvalues of A. This is already the main thing one wants from the algorithm: staring at the diagonal of A_m for big m reveals the eigenvalues of A to ever greater accuracy.

We stop the argument at this point, without addressing the stronger convergence result, since we have already explained the key ideas behind the two main points: why the maneuver of flipping the QR-products is natural (as in the analysis of A^4 above, adapted to each A^m) and why an upper triangular property emerges in the A_m's as m grows.

"... in all my life I have never labored at all as hard, and ... I have become imbued with a great respect for mathematics, the subtle parts of which, in my innocence, I had till now regarded as pure luxury."

A. Einstein

I. Newton's method for optimization (optional)

An application of the Hessian from Chapter 25 to numerical optimization of functions $f : \mathbf{R}^n \to \mathbf{R}$ is *Newton's method for optimization*. It is often much faster than gradient descent. This is really an application to $\nabla f : \mathbf{R}^n \to \mathbf{R}^n$ of what is called Newton's method in Section 18.5, but we introduce it here from scratch for the intended application (so it isn't necessary to look back in Section 18.5).

I.1. The algorithm. Recall the idea of gradient descent in Section 11.3: to optimize the function $f(x_1, \dots, x_n)$, move from an initial point $\mathbf{a}$ some distance in the direction of $\pm(\nabla f)(\mathbf{a})$. One of the major problems with this technique is that we didn't clarify *how far* to move in the direction of the gradient or its negative.

Let us revisit that situation. Starting at $\mathbf{x} = \mathbf{a}$, how should we move to try to get to a critical point of f? By definition, a critical point $\mathbf{c}$ of f satisfies $(\nabla f)(\mathbf{c}) = \mathbf{0}$. The gradient is a vector-valued function $g = \nabla f : \mathbf{R}^n \to \mathbf{R}^n$, so according to (13.5.4) applied to g we can approximate $(\nabla f)(\mathbf{a} + \mathbf{h}) = g(\mathbf{a} + \mathbf{h})$ to first order in $\mathbf{h}$ by the formula

$$(\nabla f)(\mathbf{a}+\mathbf{h}) = g(\mathbf{a}+\mathbf{h}) \approx g(\mathbf{a}) + ((Dg)(\mathbf{a}))\mathbf{h} = (\nabla f)(\mathbf{a}) + ((Dg)(\mathbf{a}))\mathbf{h}$$

for the $n \times n$ matrix $(Dg)(\mathbf{a})$.

Now comes a pleasant surprise, when we calculate exactly what $(Dg)(\mathbf{a})$ is. The ij-entry of this matrix is $\partial g_i/\partial x_j$ for the ith component function $g_i : \mathbf{R}^n \to \mathbf{R}$ of $g = \nabla f : \mathbf{R}^n \to \mathbf{R}^n$. By the definition of ∇f, its ith component function g_i is $\partial f/\partial x_i = f_{x_i}$. Hence, the ij-entry of $(Dg)(\mathbf{a})$ is $(g_i)_{x_j}(\mathbf{a}) = f_{x_i x_j}(\mathbf{a})$. This is exactly the ij-entry of the Hessian $(\mathrm{H}f)(\mathbf{a})$ of f at $\mathbf{a}$!

In other words, the linear approximation to the gradient of f at $\mathbf{a}$ has the form:

$$(\nabla f)(\mathbf{a}+\mathbf{h}) \approx (\nabla f)(\mathbf{a}) + ((\mathrm{H}f)(\mathbf{a}))\mathbf{h}.$$

We want this to equal $\mathbf{0}$ for a suitable (hopefully small) choice of $\mathbf{h}$. Setting the approximate expression on the right side equal to $\mathbf{0}$ is a linear system for the unknown $\mathbf{h}$. If the matrix $(\mathrm{H}f)(\mathbf{a})$ is invertible (as it often is), we can solve for $\mathbf{h}$ using the matrix inverse: $\mathbf{h} \approx -((\mathrm{H}f)(\mathbf{a}))^{-1}((\nabla f)(\mathbf{a}))$. Summarizing:

(Newton's method for optimization) To find a critical point of $f : \mathbf{R}^n \to \mathbf{R}$ (i.e., a zero of $\nabla f : \mathbf{R}^n \to \mathbf{R}^n$) near $\mathbf{a} \in \mathbf{R}^n$, consider the sequence of vectors $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \dots \in \mathbf{R}^n$ defined by $\mathbf{a}_1 = \mathbf{a}$ and

$$\mathbf{a}_{k+1} = \mathbf{a}_k - ((\mathrm{H}f)(\mathbf{a}_k))^{-1}((\nabla f)(\mathbf{a}_k)) \text{ for all } k \geq 1 \tag{I.1.1}$$

(provided $(\mathrm{H}f)(\mathbf{a}_k)$ is invertible for every k, as usually happens). This sequence of n-vectors often converges rapidly to a critical point of f.

Remark I.1.1. For $n = 1$, this procedure to find a zero of f' replaces a with $a - f'(a)/f''(a)$. This is the application to the function f' of the version of Newton's method given in some single-variable calculus books to find zeros of a function. See Remark 18.5.5 for further discussion and references on this case.

Remark I.1.2. When implementing Newton's method to solve high-dimensional optimization problems (such as arise in machine learning), one must use more sophisticated versions of the procedure we have just described. This is necessary to bypass practical drawbacks such as the "cost" in computer time to compute a Hessian and to store its inverse in computer memory. Another serious issue is that the method seeks critical points and so treats local maxima, local minima, and saddle points on equal footing; e.g., when $(\nabla f)(\mathbf{a}) \neq \mathbf{0}$, the projection of the "Newton step" $-((\mathrm{H}f)(\mathbf{a}))^{-1}(\nabla f)(\mathbf{a})$ into the line spanned by $(\nabla f)(\mathbf{a})$ might be in the direction of $(\nabla f)(\mathbf{a})$ rather than $-(\nabla f)(\mathbf{a})$ unless f is strictly convex (see Remark 26.1.6). The book [NW] has an extensive discussion of enhancements of Newton's method to improve its utility in a variety of applications.

Example I.1.3. We noted in Examples 11.3.4 and 18.5.4 that gradient descent is sometimes too slow for solving a minimization problem that arises in black hole imaging. One of the first techniques developed to handle this replaced a multivariable function with a quadratic approximation as in (25.3.1) (see [**BJZFDF1**, Sec. 4.3(2)]). The deviation from a critical point is estimated by solving a system of linear equations via an inverse matrix, rederiving in that setting the formula (I.1.1) in Newton's method for optimization: see equation (11) in [**BJZFDF2**, Sec. 2.2], where they denote by X what we call $\mathbf{h}$; the sign in (I.1.1) doesn't appear there because their expression for the term corresponding to $(\nabla f)(\mathbf{a})$ has an overall sign in front and so the signs cancel. Ongoing research for imaging of rapidly evolving black holes uses quadratic approximations to save a lot of time in the optimization work. ■

Example I.1.4. The need to rapidly solve non-linear equations has been discussed in Section 18.5 (for robotics and computer animation in Example 18.5.2, and for GPS in Example 18.5.3). If we formulate such a system of equations in vector language as $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ for some $\mathbf{f} : \mathbf{R}^n \to \mathbf{R}^m$ having component functions $f_1, \dots, f_m : \mathbf{R}^n \to \mathbf{R}$, note that $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ precisely when

$$0 = \|\mathbf{f}(\mathbf{x})\|^2 = f_1(\mathbf{x})^2 + \cdots + f_m(\mathbf{x})^2.$$

This leads to an interesting idea: finding a simultaneous zero to the (typically non-linear) f_j's is an instance of minimizing the function $F(\mathbf{x}) = \sum_{j=1}^{m} f_j(\mathbf{x})^2$, so maybe we can try to find such a solution via optimization techniques! It might happen in some context that we can't find a zero of $\mathbf{f}$ but the minimization problem for $\sum f_j^2$ still has a solution, and that could wind up being "good enough" for a specific practical application. This actually does arise in practice, such as for the non-linear robotics equations referenced in Example 18.5.2; see [**AL**, Sec. 9.3.3] ■

I.2. Worked examples and divergence.

Example I.2.1. Consider $f(x,y) = 5x^2 + 5y^2 - y - x^2y$. We have $(\nabla f)(x,y) = \begin{bmatrix} 10x - 2xy \\ 10y - x^2 - 1 \end{bmatrix}$, so critical points correspond to simultaneous solutions of $10x - 2xy = 0$ and $10y = 1 + x^2$. The first equation says $2x(5-y) = 0$, so either $x = 0$ or $y = 5$. Using $10y = 1 + x^2$, we get three critical points: $(0, 1/10), (7,5), (-7,5)$. One checks via the Hessian that $(0, 1/10)$ is a local minimum and $(\pm 7, 5)$ are saddle points.

Let's see what Newton's method for optimization gives in this case. The Hessian is

$$(\mathrm{H}f)(x,y) = \begin{bmatrix} 10 - 2y & -2x \\ -2x & 10 \end{bmatrix}.$$

Starting at some $\mathbf{a} \in \mathbf{R}^2$, we repeatedly replace $\mathbf{a}$ as follows: $\mathbf{a} \rightsquigarrow \mathbf{a} - ((\mathrm{H}f)(\mathbf{a}))^{-1}(\nabla f)(\mathbf{a})$.

We will start from various $\mathbf{a}$ and see what happens, to an accuracy of three decimal digits:

$$(0,0) \rightsquigarrow (0, 0.1) \rightsquigarrow (0, 0.1) \rightsquigarrow (0, 0.1)$$

$$(1,1) \rightsquigarrow (-0.263, 0.053) \rightsquigarrow (-0.008, 0.093) \rightsquigarrow (0.000, 0.099) \approx (0, 0.1)$$

$$(4,4) \rightsquigarrow (10, 6.5) \rightsquigarrow (7.628, 5.356) \rightsquigarrow (7.044, 5.027) \rightsquigarrow (7.000, 5.000)$$

$$(-10,7) \rightsquigarrow (-7.682, 5.464) \rightsquigarrow (-7.055, 5.038) \rightsquigarrow (-7.000, 5.000)$$

In each case, the process rapidly finds (at least to three decimal digits of accuracy) a critical point, though sometimes it converges to a saddle point rather than to a local extremum. ■

There is an issue with Newton's method for optimization in general that doesn't arise in the preceding example: it is quite possible for the iteration of Newton's method to diverge! That is, rather than approaching a critical point, the coordinates of the points may get larger and larger (or wander around without settling down). So just as with gradient descent, one needs to be careful about where to start: if one starts *close enough* to a critical point then Newton's method behaves well, but in general it can behave very badly. If we test Newton's method for optimization on some more complicated examples we have encountered, such as Example 11.3.6, then the divergence possibility really can occur, as we now show.

Example I.2.2. Let's try to find a critical point of the function

$$f(x,y) = x + y - (0.1)(\ln(x) + \ln(y) + \ln(2x+3y-1) + \ln(3x+y-1))$$

near $(x, y) = (1, 1)$. In Example 11.3.6 we computed $(\nabla f)(x,y) = \begin{bmatrix} 1-(0.1)\left(\frac{1}{x}+\frac{2}{2x+3y-1}+\frac{3}{3x+y-1}\right) \\ 1-(0.1)\left(\frac{1}{y}+\frac{3}{2x+3y-1}+\frac{1}{3x+y-1}\right)\end{bmatrix}$.
From this we can compute the Hessian symbolically:

$$(\mathrm{H}f)(x,y) = (0.1)\begin{bmatrix} \frac{1}{x^2}+\frac{4}{(2x+3y-1)^2}+\frac{9}{(3x+y-1)^2} & \frac{6}{(2x+3y-1)^2}+\frac{3}{(3x+y-1)^2} \\ \frac{6}{(2x+3y-1)^2}+\frac{3}{(3x+y-1)^2} & \frac{1}{y^2}+\frac{9}{(2x+3y-1)^2}+\frac{1}{(3x+y-1)^2}\end{bmatrix}.$$

This is a mess, but it doesn't matter because your computer will now get to have all the fun. Starting with some $\mathbf{a}$, the procedure is to replace $\mathbf{a}$ as follows: $\mathbf{a} \rightsquigarrow \mathbf{a} - ((\mathrm{H}f)(\mathbf{a}))^{-1}(\nabla f)(\mathbf{a})$. If we start with $\mathbf{a} = \begin{bmatrix}1/3\\1/3\end{bmatrix}$, we get the following (which converges rather well in each vector entry):

$$\begin{aligned}\begin{bmatrix}1/3\\1/3\end{bmatrix} &\rightsquigarrow \begin{bmatrix}0.4129979035639412997903563941299790356\,4\ldots\\0.2620545073375262054507337526205450733\,8\ldots\end{bmatrix}\\ &\rightsquigarrow \begin{bmatrix}0.448503604285796062169000750331503091\,03\ldots\\0.255469419018950445643281434422097365\,80\ldots\end{bmatrix}\\ &\rightsquigarrow \begin{bmatrix}0.455262908136211701394916829750668269\,14\ldots\\0.254225854645430719087532887362933893\,16\ldots\end{bmatrix}\\ &\rightsquigarrow \begin{bmatrix}0.455440619419144083608368768117483558\,10\ldots\\0.254197950922193951537819255009859951\,00\ldots\end{bmatrix}\\ &\rightsquigarrow \begin{bmatrix}0.455440737979187439435823225674108403\,54\ldots\\0.254197933805846048297285754313467731\,07\ldots\end{bmatrix}.\end{aligned}$$

Note that to three decimal digits' accuracy, in *three* steps this reached *the same critical point as we found in Example 11.3.6 using 100 steps of gradient descent.* Moreover, at the point we just reached in five steps, ∇f vanishes to 12 decimal digits in each entry (it is very close to a critical point of f!).

But how did we know to start with $\mathbf{a} = \begin{bmatrix}1/3\\1/3\end{bmatrix}$? What if, say, we had started with $\mathbf{a} = \begin{bmatrix}1\\1\end{bmatrix}$? Then we would get this: $\begin{bmatrix}1\\1\end{bmatrix} \rightsquigarrow \begin{bmatrix}-1.128\\-2.830\end{bmatrix} \rightsquigarrow \begin{bmatrix}-7.811\\-50.938\end{bmatrix} \rightsquigarrow \begin{bmatrix}-105.724\\-11455.719\end{bmatrix} \rightsquigarrow$ something really big. Disaster! The vector $\mathbf{a}$ gets bigger at each step, not going towards anything. We hit a real problem with Newton iteration: if you don't start close enough to a critical point, you can get nonsense. This is a very interesting issue to understand, but it is outside the scope of this course to discuss further here. ■

Example I.2.3. For $f(x,y) = 8x^2 + 6xy + 4x^3 + 3xy^2$ we have

$$\nabla f = \begin{bmatrix} 16x + 6y + 12x^2 + 3y^2 \\ 6x + 6xy \end{bmatrix}, \quad \mathrm{H}f = \begin{bmatrix} 16 + 24x & 6 + 6y \\ 6 + 6y & 6x \end{bmatrix}.$$

(As practice, please work out the gradient and Hessian for yourself to confirm these expressions.)

Let's first show that the critical points of f are $(0,0)$, $(0,-2)$, $(-3/2,-1)$, $(1/6,-1)$. The critical points (a,b) are points at which both f_x and f_y vanish. From the formulas as given in the expression for ∇f, we need to simultaneously solve

$$16a + 6b + 12a^2 + 3b^2 = 0, \quad 6a + 6ab = 0.$$

The left side of the second equation factors as $6a(1+b)$, so it vanishes precisely when $a = 0$ or $b = -1$. We substitute each option into the first equation to see what happens.

With $a = 0$, the first equation becomes $6b + 3b^2 = 0$, for which the left side factors as $3b(2+b)$. This vanishes exactly for $b = 0$ and $b = -2$, yielding the critical points $(0,0)$ and $(0,-2)$. With $b = -1$, the first equation becomes $16a - 6 + 12a^2 + 3 = 0$, or equivalently $12a^2 + 16a - 3 = 0$. By the quadratic formula, this has as its solutions

$$a = \frac{-16 \pm \sqrt{256 + 144}}{24} = \frac{-16 \pm \sqrt{400}}{24} = \frac{-16 \pm 20}{24} = \frac{4}{24} \text{ and } \frac{-36}{24}.$$

These two options simplify to $1/6$ and $-3/2$, yielding the critical points $(-3/2,-1)$ and $(1/6,-1)$. Using the Hessian, one can check that $(0,0)$ and $(0,-2)$ are saddle points, $(-3/2,-1)$ is a local maximum, and $(1/6,-1)$ is a local minimum.

As an illustration of Newton's method for optimization, next we carry out one step of the method at the following points (and see how the output approaches some of the critical points we have found):

- $(0,-1/2)$ (near the critical points $(1/6,-1)$ and $(0,0)$, slightly closer to $(0,0)$),
- $(-1,-1)$ (near the critical point $(-3/2,-1)$),
- $(0,1)$ (closest to the critical point $(0,0)$, though not especially close to it).

Warning. If one begins at $(a,-1)$ with $a = (-8 \pm \sqrt{91})/18 \approx 0.0855, -0.9744$ then the Hessian is diagonal with nonzero diagonal entries, so it is invertible, but the output of the first step is $(0,-1)$ at which the Hessian is not invertible. Newton's method thereby gets stuck after one step! Avoiding this type of problem is an issue one has to grapple with when implementing Newton's method.

In each case, if the initial point is denoted $\mathbf{a}$ then the point we seek to calculate for the output of the first step of Newton's method for optimization is

$$\mathbf{a} - ((\mathrm{H}f)(\mathbf{a}))^{\ 1}(\nabla f)(\mathbf{a}).$$

So we aim to evaluate this expression for each of the initial points $\mathbf{a} = (0,-1/2), (-1,-1), (0,1)$. For each case, first we will compute the gradient $(\nabla f)(\mathbf{a})$ and Hessian $(\mathrm{H}f)(\mathbf{a})$ via numerical evaluation at $\mathbf{a}$ of the general symbolic expressions given for ∇f and $\mathrm{H}f$. Then we will plug those into the expression of interest and carry out its complete calculation to get an exact answer.

First, suppose $\mathbf{a} = (0,-1/2)$. For this case, we compute

$$(\nabla f)(0,-1/2) = \begin{bmatrix} -6/2 + 3/4 \\ 0 \end{bmatrix} = \begin{bmatrix} -9/4 \\ 0 \end{bmatrix}, \quad (\mathrm{H}f)(0,-1/2) = \begin{bmatrix} 16 & 6-3 \\ 6-3 & 0 \end{bmatrix} = \begin{bmatrix} 16 & 3 \\ 3 & 0 \end{bmatrix}.$$

Hence, the first step of Newton's method gives the point

$$\begin{bmatrix} 0 \\ -1/2 \end{bmatrix} - \begin{bmatrix} 16 & 3 \\ 3 & 0 \end{bmatrix}^{\ 1} \begin{bmatrix} -9/4 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -1/2 \end{bmatrix} - \begin{bmatrix} 0 & 1/3 \\ 1/3 & -16/9 \end{bmatrix} \begin{bmatrix} -9/4 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -1/2 \end{bmatrix} - \begin{bmatrix} 0 \\ -3/4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1/4 \end{bmatrix}$$

that is closest to the critical point $(0,0)$ (much more so than was the initial point $(0,-1/2)$).

Next, suppose $\mathbf{a} = (-1, -1)$. For this case, we compute

$$(\nabla f)(-1,-1) = \begin{bmatrix} -16-6+12+3 \\ -6+6 \end{bmatrix} = \begin{bmatrix} -7 \\ 0 \end{bmatrix}, \quad (\mathrm{H}f)(-1,-1) = \begin{bmatrix} 16-24 & 0 \\ 0 & -6 \end{bmatrix} = \begin{bmatrix} -8 & 0 \\ 0 & -6 \end{bmatrix}.$$

Hence, the first step of Newton's method gives the point

$$\begin{bmatrix} -1 \\ -1 \end{bmatrix} - \begin{bmatrix} -8 & 0 \\ 0 & -6 \end{bmatrix}^{1} \begin{bmatrix} -7 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} - \begin{bmatrix} -1/8 & 0 \\ 0 & -1/6 \end{bmatrix} \begin{bmatrix} -7 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} - \begin{bmatrix} 7/8 \\ 0 \end{bmatrix} = \begin{bmatrix} -15/8 \\ -1 \end{bmatrix}$$

that is closest to the critical point $\begin{bmatrix} -3/2 \\ -1 \end{bmatrix}$ and slightly closer to it (by $1/8$) than is the initial point $(-1, -1)$.

Finally, suppose $\mathbf{a} = (0, 1)$. For this case we compute

$$(\nabla f)(0,1) = \begin{bmatrix} 6+3 \\ 0 \end{bmatrix} = \begin{bmatrix} 9 \\ 0 \end{bmatrix}, \quad (\mathrm{H}f)(0,1) = \begin{bmatrix} 16 & 12 \\ 12 & 0 \end{bmatrix}.$$

Hence, the first step of Newton's method gives the point

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 16 & 12 \\ 12 & 0 \end{bmatrix}^{1} \begin{bmatrix} 9 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 & 1/12 \\ 1/12 & -1/9 \end{bmatrix} \begin{bmatrix} 9 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 3/4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1/4 \end{bmatrix},$$

which is the same as we got from the first initial point we tried (and is closest to the critical point $(0, 0)$, much more so than is the initial point $(0, 1)$).

We can push the preceding computations a bit further, as follows. If you do a general symbolic calculation by hand (it is not so bad) you can check that any initial point $(x, -1)$ is taken by Newton's method to the point $(g(x), -1)$ with

$$g(x) = \frac{x}{2} - \frac{1}{3} + \frac{25}{24(2+3x)}$$

(e.g., $g(-1) = -1/2 - 1/3 - 25/24 = -45/24 = -15/8$, recovering the output of the second initial point above) and that any initial point $(0, y)$ is taken by Newton's method to the point $(0, h(y))$ with

$$h(y) = \frac{y}{2} - \frac{1}{2} + \frac{1}{2y+2}$$

(e.g., $h(-1/2) = -1/4 - 1/2 + 1/1 = 1/4$ and $h(1) = 1/2 - 1/2 + 1/4 = 1/4$, recovering the outputs of the first and third initial points above). So given a point of either type, which includes each of the ones in Example I.2.3, to run Newton's method repeatedly amounts to feeding g or h into itself.

Beginning with the output $(-15/8, -1)$ from $(-1, -1)$, running Newton's method 3 more times gives the successive outputs $(g(-15/8), 1) \approx (-1.558, -1)$, $(g(g(-15/8)), 1) \approx (-1.502, -1)$, and $(g(g(g(-15/8))), -1) \approx (-1.500002, -1)$. So after 4 steps from the initial point $(-1, -1)$, we are approximating the local maximum $(-3/2, -1)$ accurately to 5 decimal digits.

Beginning with the output $(0, 1/4)$ from $(0, -1/2)$ and $(0, 1)$, running Newton's method 3 more times gives the successive outputs $(0, h(1/4)) \approx (0, 0.025)$, $(0, h(h(1/4))) \approx (0, 0.0003)$, and

$$(0, h(h(h(1/4)))) \approx (0, 0.00000005).$$

So after 4 steps from the initial points $(0, -1/2)$ and $(0, 1)$, we are approximating the saddle point $(0, 0)$ accurately to 7 decimal digits. ■

"... as time goes on, it becomes increasingly evident that the rules which the mathematician finds interesting are the same as those which Nature has chosen."

P. Dirac, co-winner of 1933 Nobel Prize in Physics

J. Hessians and chemistry (optional)

Knowledge about the definiteness properties of the Hessian of $f : \mathbf{R}^n \to \mathbf{R}$ at a critical point (e.g., positive-definite, indefinite, etc.) is a valuable tool in the study of molecular structure. In this setting, the function f is the *electronic energy* E_e of a molecule; this is a function of an n-vector $\mathbf{x}$ whose entries record the position in $\mathbf{R}^3$ of every atom in a molecule (so even for molecules with 5 or 6 atoms, n is somewhat large, around 15 or more). The $\mathbf{R}^n$ in which $\mathbf{x}$ lives is called the "state space" since it keeps track of the state of the molecular structure, and one then calls $\mathbf{x}$ the "state vector". Finding local minima of the function $E_e(\mathbf{x})$ is rather useful, as we now illustrate in two ways via the Hessian.

J.1. Positive-definite Hessian and molecular structure. At a given temperature, the local minima for the electronic energy function E_e of a single molecule correspond to the possible molecular structures. The state vector $\mathbf{x}$ tends to wander around near the local minimum corresponding to whatever is the current molecular structure. To find the locations of the local minima, chemists use a computer program that encodes the necessary molecular physics to calculate $E_e(\mathbf{x})$ and its gradient for an initial choice of $\mathbf{x}$. A chemist then carries out gradient descent (Section 11.3) until reaching an $\mathbf{a}$ at which the gradient vanishes (numerically). This is a critical point, but it might not be a local minimum. If the Hessian of E_e at this critical point $\mathbf{a}$ has all positive eigenvalues then that confirms that a genuine local minimum has been found. Other initial choices of $\mathbf{x}$ can be tried, which, if sufficiently different from the first choice, may travel under gradient descent to other local minima $\mathbf{b}$ and thereby correspond to other possible molecular structures.

As the atoms vibrate and twist (due to the surrounding temperature), $\mathbf{x}$ changes and correspondingly the value of the energy $E_e(\mathbf{x})$ changes. When the molecule is in a local minimum state $\mathbf{a}$ for E_e at which the Hessian has positive eigenvalues $\lambda_1, \dots, \lambda_n$, a lot of quantitative information is encoded in the λ_i's: they allow chemists to calculate frequencies seen in vibrational spectroscopy for the corresponding molecular structure, and to estimate the amount of heat stored in the molecule as well as its entropy and its free energy as the temperature (and therefore the amount of atomic motion) increases.

J.2. Indefinite Hessian and molecular transition. The critical points of E_e with an *indefinite* Hessian help for describing transitions between different molecular structures. We now illustrate this in the setting of Figure J.2.1, which gives two ways of double-bonding collections of hydrogen and carbon atoms, called cis-2-butene and trans-2-butene.

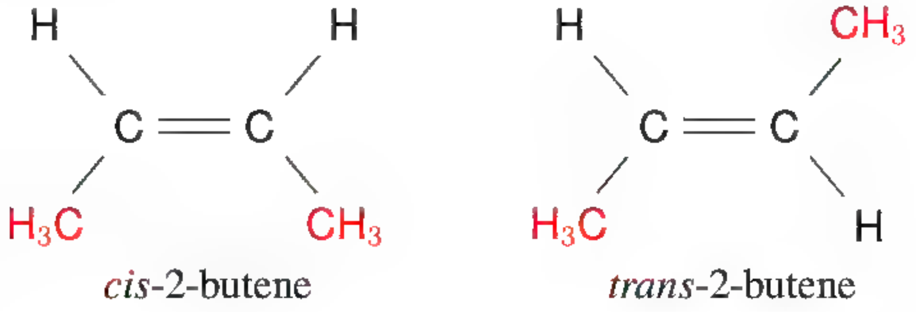


FIGURE J.2.1. Two molecules whose transition involves a critical point with one negative eigenvalue for the Hessian of electronic energy

At a given temperature the two molecular states in Figure J.2.1 correspond to two different local minima $\mathbf{a}$ (for cis-2-butene) and $\mathbf{b}$ (for trans-2-butene) of the electronic energy function E_e, with $E_e(\mathbf{b})$ a bit smaller than $E_e(\mathbf{a})$. Since both $\mathbf{a}$ and $\mathbf{b}$ are basins for E_e (they are local minima), for the molecule to transition from one state to the other it needs to acquire some more energy to escape the basin it is in and to fall into another basin.

The transition is rare: at a temperature as high as 500° Celsius it usually takes 3 minutes for such a transition to occur, and at 450° Celsius it takes around 2 days for such a transition to occur [LL] (and at room temperature it takes an incredibly long time). The reason for this rarity is due to the presence of an indefinite critical point $\mathbf{c}$ for E_e halfway between $\mathbf{a}$ and $\mathbf{b}$ (in the state space), whose energy level $E_e(\mathbf{c})$ exceeds $E_e(\mathbf{a})$ and $E_e(\mathbf{b})$ by a large amount and for which the Hessian of E_e at $\mathbf{c}$ has *only one* negative eigenvalue (the others are all positive). The visualization you should keep in mind for the discussion below with $\mathbf{c}$ is that of a saddle point high up above two different basins and a ball in one basin bouncing around and being given random energy boosts. Only rarely will the ball have enough energy to reach the level of the high-up saddle point, and when the ball does so it tends to drop into the basin on the other side provided it is *moving in the right direction*. The bouncing ball plays the role of the state vector $\mathbf{x}$ and the saddle point plays the role of $\mathbf{c}$.

If we let L be the line in the state space for the only negative eigenvalue of the Hessian of E_e at $\mathbf{c}$ and decompose the state vector $\mathbf{x}$ into its components along L and the $(n-1)$-dimensional $L^{\perp}$, for $\mathbf{x}$ near $\mathbf{c}$ the positivity of all but one of the eigenvalues tends to keep $\mathbf{Proj}_{L^{\perp}}(\mathbf{x})$ constrained to be near $\mathbf{c}$ (much like being in a basin). To transition from $\mathbf{a}$ to $\mathbf{b}$ requires that L-component of the motion of $\mathbf{x}$ near $\mathbf{a}$ gets enough of a kick to increase $E_e(\mathbf{x})$ to exceed $E_e(\mathbf{c})$ and to be moving in the right direction to go over the saddle at $\mathbf{c}$ and drop down to the other side and land in basin $\mathbf{b}$. In particular, the transition from $\mathbf{a}$ to $\mathbf{b}$ generally always goes through the intermediate state corresponding to $\mathbf{c}$, and that happens rarely because of the big difference between $E_e(\mathbf{a})$ and $E_e(\mathbf{c})$.

Remark J.2.1. The same qualitative reasoning as in the preceding example applies to the transition from $\mathbf{b}$ to $\mathbf{a}$ (also going through $\mathbf{c}$). Hence, in the presence of *many* molecules of one or both types the overall process of transition in both directions among the molecules is a 2-state Markov chain as in Chapter 16, with different probabilities for transitioning in each direction. Studying the behavior of large powers of the corresponding 2×2 Markov matrix allows chemists to explain the long-term behavior of such a system, such as why (at a given temperature) the two molecular structures are found in definite proportions after a long time has passed and what those proportions are; this is controlled by the eigenvalues of that Markov matrix. The topic of large powers of such matrices is discussed in Section 27.2, using our knowledge of eigenvalues.

References

[Al] C. Alexander, *Market Risk Analysis: Quantitative Methods in Finance*, John Wiley & Sons, West Sussex, 2008.

[ABB] O. Alter, P.O. Brown, D. Botstein, “Singular value decomposition for genome-wide expression data processing and modeling”, Proceedings of the National Academy of Sciences (USA), Vol. 97, No. 18, (2000), pp. 10101-10106.

[AL] A. Antoniou, W-S. Lu, *Practical Optimization: Algorithms and Engineering Applications*, Springer-Verlag, New York , 2007.

[Apol] Apollonius of Perga, *Treatise on conic sections* (T.L. Heath ed.), Cambridge Univ. Press, Cambridge, 1896.

[Ap] T. Apostol, *Calculus* Vol. II (2nd ed.), Wiley & Sons, New York, 1969.

[Arch] Archimedes, *The Works of Archimedes* (T.L. Heath ed.), Cambridge Univ. Press, Cambridge, 1897.

[Aru1] F. Arute et al., “Quantum supremacy using a programmable superconducting processor”, Nature, Vol. 574, 2019, pp. 505-510.

[Aru2] F. Arute et al., Suppplement to “Quantum supremacy using a programmable superconducting processor”, Nature, Vol. 574, 2019.

[Aul] K. Auletta, “Outside the Box”, The New Yorker (Feb. 3, 2014), pp. 54-61.

[BG] D.P. Bertsekas, R.G. Gallager, *Data Networks* (2nd ed.), Prentice Hall, Englewood Cliffs, 1992.

[BSS] P.A. van Bijlert, A.J. van Soest, A.S. Schulp, “Natural Frequency Method: estimating the preferred walking speed of *Tyrannosaurus rex* based on tail natural frequency”, Royal Society Open Science, Vol. 8, No. 4 (2021).

[CBJD] M. Caron, P. Bojanowski, A. Joulin, M. Douze, “Deep Clustering for Unsupervised Learning”, European Conference on Computer Vision, Lecture Notes in Computer Science, Vol. 11218, Springer-Verlag (2018), pp. 139-156.

[BJZFDF1] K.L. Bouman, M.D. Johnson, D. Zoran, V.L. Fish, S.S. Doeleman, W.T. Freeman, “Computational Imaging for VLBI Reconstruction”, Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 913-922.

[BJZFDF2] K.L. Bouman, M.D. Johnson, D. Zoran, V.L. Fish, S.S. Doeleman, W.T. Freeman, “Computational Imaging for VLBI Reconstruction: Supplemental Material”.

[Bou] N. Bourbaki, *Lie Groups and Lie Algebras* (Ch. IV-VI), Springer-Verlag, New York, 2002.

[BV] S. Boyd, L. Vandenberghe, *Introduction to Applied Linear Algebra*, Cambridge University Press, 2018.

[BP] S. Brin, L. Page, “The anatomy of a large-scale hypertextual Web search”, Proc. 7th World Wide Web Conf., 1998, pp. 107-117.

[BS] M. Brin, G. Stuck, *Introduction to Dynamical Systems*, Cambridge University Press, Cambridge, 2002.

[BSt] R. Bulirsch, J. Stoer, *Introduction to Numerical Analysis* (3rd ed.), Springer-Verlag, New York, 2002.

[BSH] A. Bureau, S. Shiboski, J.P. Hughes, “Applications of continuous time hidden Markov models to the study of misclassified disease outcomes”, Statistics in Medicine, Vol. 22, No. 3 (2003), pp. 441-462.

[CMP] L. Luca Cavalli-Sforza, P. Menozzi, A. Piazza, *The History and Geography of Human Genes*, Princeton University Press, Princeton, 1994 (1032 pp.).

[CT] L. Chang, D.Y. Tsao, “The Code for Facial Identity in the Primate Brain”, Cell, Vol. 169 (2017), pp. 1013-1028.

[CFHSSY] R. Chetty, J.N. Friedman, N. Hilger, E. Saez, D.W. Schanzenbach, D. Yagan, “How does your kindergarten classroom affect your earnings? Evidence from Project Star”, The Quarterly Journal of Economics, Oxford University Press, Vol. 126, No. 4 (Nov., 2011), pp. 1593-1660.

[CDK] L.O. Chua, C.A. Desoer, E.S. Kuh, *Linear and Nonlinear Circuits*, McGraw-Hill, New York, 1987.

[C] Y. Choquet-Bruhat, “Some memories from meeting Einstein in 1951-1952”, 2015.

[Coh] H. Cohen, *Number Theory Volume II: Analytic and Modern Tools*, Springer-Verlag, New York, 2007.

[Col] J. Colt, *Computation of Dissolved Gas Concentration in Water as Functions of Temperature, Salinity and Pressure* (2nd ed.), Elsevier, 2012.

[Daj] L. Dajose, “Cracking the Code of Facial Recognition”, June 2017.

[Dan] G.P. Dandelin, “Mémoire sur quelques propriétés remarquables de la focale parabolique”, Nouveaux mémories de l’Académie royale de Bruxelles, Vol. 2 (1822), pp. 171-200.

[DKMSZ] V. Dani, V. King, M. Movahedi, J. Saia, M. Zamani, “Secure multi-party computation in large networks”, Distributed Computing, Vol. 30, No. 3 (2017), pp. 193-229.

[Dan] G. Dantzig, "The Diet Problem", Interfaces, Vol. 20, No. 4 (1990), pp. 43-47.

[DM] R. Davidson, J. MacKinnon, *Econometric Theory and Methods*, Oxford Univ. Press, Oxford, 2003.

[DM] E. Davis, G. Marcus, "Are Neural Networks About to Reinvent Physics?", Nautilus, Vol. 78 (Ch. 3), 2019.

[DS] J. Dongarra, F. Sullivan, "Top Ten Algorithms of the Century", Computing in Science & Engineering, Vol. 2, No. 01 (2000), pp. 22-23.

[Do] D. Donoho, "From Blackboard to Bedside: high-dimensional geometry is transforming the MRI industry", Congressional Briefing (June 28, 2017).

[Dry] H. Drygas, "On the relationship between the method of least squares and Gram–Schmidt orthogonalization", Acta et Commentationes Universitatis Tartuensis de Mathematica, Vol. 15, No. 1 (2011).

[Du] M. Duchin, "Gerrymandering metrics: How to measure? What's the baseline?", Bulletin of the American Academy of Arts & Sciences, Vol. LXXI, No. 2 (Winter 2018), pp. 54-58.

[En] S. B. Engelsman, *Families of Curves and the Origins of Partial Differentiation*, Mathematical Studies **93**, North-Holland, Amsterdam, 1984.

[FF] Q. Feng, P. Frazier, "Gleaning Insights from Uber's Partner Activity Matrix with Genomic Biclustering and Machine Learning" (2017).

[Feyn1] R.P. Feynman, R. Leighton, M. Sands, *Feynman Lectures on Physics*, Addison-Wesley, Reading MA, 1963-5.

[Feyn2] R.P. Feynman, "The Development of the Space-Time View of Quantum Electrodynamics", Nobel Lectures, Physics 1963-1970, Elsevier, Amsterdam (1972).

[FS] D. Frenkel, B. Smit, *Understanding molecular simulation: from algorithms to applications* (2nd ed.), Computational Science Series, Vol. 1, Academic Press, Sam Diego, 2002.

[Gilb] G. Gilbert, "Positive Definite Matrices and Sylvester's Criterion", American Math. Monthly, Vol. 98, No. 1 (1991), pp. 44-46.

[Gill] J. Gill, *Bayesian Methods: A Social and Behavioral Sciences Approach* (2nd ed.), Chapman & Hall, London, 2008.

[GU] G. Golub, F. Uhlig, "The QR Algorithm: 50 years later – its genesis by John Francis and Vera Kublanovskaya and subsequent developments", IMA Journal of Numerical Analysis, Vol. 29 (2009), pp. 467-485.

[HB] J.A. Harris, F.G. Benedict, "A biometric study of human basal metabolism", Proceedings of the National Academy of Sciences (USA), Vol. 4, No. 2 (1918), pp. 370-373.

[Ha1] T. Hawkins, "The Theory of Matrices in the 19th Century", Proceedings of the International Congress of Mathematicians, Canadian Mathematical Congress, Vol. 2 (1975), pp. 561-570.

[Ha2] T. Hawkins, "Continued fractions and the origins of the Perron–Frobenius theorem", Arch. Hist. Exact Sci. , Vol. 62 (2008), pp. 655-717.

[Ha3] T. Hawkins, *The Mathematics of Frobenius in Context*, Springer-Verlag, New York, 2013.

[He] D. Heaven, "Why deep-learning AI's are so easy to fool", Nature, Vol. 574 (2019), pp. 163-166.

[Hig] N. Higham, "SIAG/LA Prize Winners Speed Up the QR Algorithm", SIAM News, Vol. 36 (2003).

[Hil] D. Hilbert, "Naturerkennen und Logik", Die Naturwissenschaften, Vol. 18 (1930), pp. 959-963.

[Hor] H.J. ter Horst, "Fundamental Functions in Equilibrium Thermodynamics", Annals of Physics Vol. 176 (1987), pp. 183-217.

[Hou] A.S. Householder, "Unitary Triangularization of a Non-symmetric Matrix", Journal of the Association for Computing Machinery, Vol. 5, No. 4 (1958), pp. 339-342.

[Hsu] J. Hsu, "Artificial Intellligence Could Improve Health Care for All – Unless it Doesn't", Undark (June, 2019).

[KK] F. Kachapova, I. Kachapov, "Orthogonal projection in teaching regression and financial mathematics", Journal of Statistics Education, Vol. 18, No. 1 (2010), pp. 1-18.

[KW] W. Kamakura, M. Wedel, *Market Segmentation: Conceptual and Methodological Foundations* (2nd ed.), Kluwer Academic Publishers, Norwell MA, 2000.

[KH] E.D. Kaplan, C.J. Hegarty, *Understanding GPS/GNSS: Principles and Applications* (3rd ed.), Artech House, Boston, 2016.

[KT] S. Karlin, H. Taylor, *An Introduction to Stochastic Modeling*, Academic Press, Orlando, 1984.

[KMS] M. Kato, G. Matsumoto, T. Sakai, "A Note on the Hawkins-Simon Conditions", Hokudai Economics Papers Vol. 3 (1972), pp. 46-48.

[KP] B.,Kaufman, E.C. Williams, C. Underkoffler, R. Pederson, N. Mardirossian, I. Watson, J. Parkhill, "COATI: Multi-model Contrastive Pretraining for Representing and Traversing Chemical Space", Journal of Chemical Information and Modeling, Vol. 64, No. 4 (2024), pp. 1145-1157.

[Kn] D. Knuth, *The Art of Computer Programming*, Vol. I (3rd ed.), Addison-Wesley, New York, 1997.

[KS] J. Kocik, A. Solecki, "Disentangling a Triangle", Amer. Math. Montly, Vol. 116, No. 3 (March, 2009), pp. 228-237.

[Kr] W. Krauth, *Statistical mechanics: Algorithms and Computations*, Oxford Master Series in Statistical, Computational, and Theoretical Physics, Oxford University Press, Oxford, 2006.

[Kü] T. Kühn, *Eigenvalues of integral operators generated by positive-definite Höolder kernels on metric compacta*, Indag. Math, Vol. 49 (1987), pp. 51-61.

[LX] T.L. Lai, H. Xing, *Statistical Models and Methods for Financial Markets*, Springer-Verlag, New York, 2008.

[LMT] M-S. Lee, G. Medioni, C-K. Tang, "Tensor Voting: Theory and Applications", Proceedings of the Reconnaissance des Formes et Intelligence Artificielle, Paris, 2000.

[Le] S.D. Levitt, "Using repeat challengers to estimate the effect of campaign spending on election outcomes in the U.S. House", Journal of Political Economy, Vol. 102, No. 4 (Aug.,1994), pp. 777-798.

[Li] P.W. Likins, "Effects of Energy Dissipation on the Free Body Motions of Spacecraft", JPL Technical Report No. 32-860 (July, 1966).

[LL] M. Lin, K.J. Laidler, "Some Aspects of the Thermal cis-trans Isomerization Mechanisms", Canadian Journal of Chemistry, Vol. 46, No. 6 (1968), pp. 973-978.

[MO] S. Machup, L. Onsager, "Fluctuations and Irreversible Processes II. Systems with Kinetic Energy", Physical Review, Vol. 91, No. 6 (Sept., 1953), pp. 1512-1515.

[MP] C. Martin, M. Porter, "The Extraordinary SVD", American Math. Monthly, Vol. 119 (Dec. 2012), pp. 838-851.

[Max] J.C. Maxwell, "Address to the Mathematical and Physical Sections of the British Association", in *Scientific Papers of James Clerk Maxwell* (Vol. II), Dover Publications, New York, 1965.

[M] C. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2000.

[MB] R. Miller, P. Blair, *Input-Output Analysis: Foundations and Extensions* (2nd ed.), Cambridge Univ. Press, 2009.

[Neu] J. von Neumann, *Mathematical Foundations of Quantum Mechanics*, Princeton Univ. Press, Princeton, 1932.

[NW] J. Nocedal, S.J. Wright, *Numerical Optimization* (2nd ed.), Springer-Verlag, New York, 2006.

[N] J. Novembre, T. Johnson, K. Bryc, Z. Kutalik, A.R. Boyko, A. Auton, A. Indap, K.S. King, S. Bergmann, M.R. Nelson, "Genes mirror geography within Europe", Nature, Vol. 456 (2008), pp. 98-101.

[Pa] B. Parlett, Review of "The Algebraic Eigenvalue Problem", SIAM Review, Vol. 8, No. 4 (1965), pp. 543–5.

[Pe] O. Perron, *Die Lehre von den Kettenbrüchen*, Bd. I-II (3te Aufl.), B.G. Teubner Verlagsgesellschaft, Stuttgart, 1957.

[Pi] J. Pinsker, "The financial perks of being tall", Atlantic Monthly (May, 2015).

[Pu] M.L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley, Hoboken, 1994.

[Q] R.Q. Quiroga, "How do we Recognize a Face?", Cell, Vol. 169 (2017), pp. 975-976.

[RPGS] S.C. Rambaud, J.G. Pérez, M.A.S. Granero, J.E.T. Segovia, "Markowitz's model with Euclidean vector spaces", European Journal of Operations Research, Vol. 196 (2009), pp. 1245-1248.

[Ri] P. Riley, "Three pitfalls to avoid in machine learning", Nature, Vol. 572 (2019), pp. 27-29.

[Ro] J.K. Rowling, *Harry Potter and the Deathly Hallows*, Scholastic, New York, 2007.

[Sam] P. Samuelson, *Foundations of Economic Analysis*, Harvard University Press, Cambridge, 1947.

[Sch1] E. Schrödinger, "Quantisierung als Eigenwertproblem I", Annalen der Physik, Vol. 79 (1926), pp. 361-376.

[Sch2] E. Schrödinger, "Quantisierung als Eigenwertproblem II', Annalen der Physik, Vol. 79 (1926), pp. 489-527.

[Sch3] E. Schrödinger, "Quantisierung als Eigenwertproblem III", Annalen der Physik, Vol. 80 (1926), pp. 437-490.

[Sch4] E. Schrödinger, "Quantisierung als Eigenwertproblem IV", Annalen der Physik, Vol. 81 (1926), pp. 109-139.

[Sch5] E. Schrödinger, "On the Relation between the Quantum Mechanics of Heisenberg, Born, Jordan, and that of Schrödinger", Annalen der Physik, Vol. 79 (1926), pp. 45-61.

[Seg] C. Segre, "On some tendencies in geometric investigations", Bulletin of the American Mathematical Society, Vol. 10, No. 9 (1904), pp. 442-468.

[Sham] A. Shamir, "How to share a secret", Communications of the ACM, Vol. 22, No. 11 (1979), pp. 612-613.

[Shang] K. Shang, "Applying Image Recognition to Insurance", Society of Actuaries Research, 2018.

[Shan] C.E. Shannon, "A Mathematical Theory of Communication", The Bell System Technical Journal, Vol. 27, No.2-3 (July, Oct. 1948), pp. 379-423, 623-656.

[Shan1] C.E. Shannon, "A Chess-Playing Machine", Scientific American, Vol. 182, No. 2 (1950), pp. 48-51.

[Shan2] C.E. Shannon, "Creative Thinking", lecture at Bell Labs (March, 1952).

[SH] D. Silver, D. Hassabis, et al., "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play", Science, Vol. 362 (2018), pp. 1140-1144.

[Steen] L.A. Steen, "Highlights in the History of Spectral Theory", American Math. Monthly, Vol. 80, No. 4 (1973), pp. 359-381.

[St] B. Steenbarger, "The Growing Crisis in Modern Finance", Forbes (May 25, 2018).

[Ta] J. Tapson, "Overfit for Purpose: Why Crowdsourced AI May Not Work for Hedge Funds", Emerj (Dec. 13, 2018).

[THW] G. Thomas, J. Hass, M. Weir, *Thomas' Calculus: Early Transcendentals, Single Variable* (13th ed.), Pearson, New York, 2014.

[Th] E. Thorp, "A perspective on quantitative finance models for beating the market", Quantitative Finance Review (2003), pp. 33-38.

[Tr] C. Truesdell, "Cauchy and the modern mechanics of continua", Revue d'Histoire des Sciences, Vol. 45, No. 1 (1992), pp. 5-24.

[T] D.Y. Tsao, "Face Value", Scientific American, Vol. 320, No. 2 (Feb., 2019), pp. 23-29.

[Tsh] V. Tshitoyan, J. Dagdelen, et al., "Unsupervisied word embeddings capture latent knowledge from materials science literature", Nature, Vol. 571, No. 7763 (2019), pp. 95-98.

[Vig] T. Vigen, *Spurious Correlations*, Hachette Books, New York, 2015.

[Wal] B. Walsh, "The Scarcity of Cross Products on Euclidean Spaces", American Math. Monthly, Vol. 74, No. 4 (1967), pp. 188-194.

[Wat] D. Watkins, "Understanding the QR Algorithm", SIAM Review, Vol. 24, No. 4 (1982), pp. 427-440.

[Wig] E. Wigner, "The unreasonable effectiveness of mathematics in the natural sciences", Communications in Pure and Applied Mathematics, Vol. XIII, No. 1 (1960), pp. 1-14.

[Wil1] J.H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

[Wil2] J.H. Wilkinson, "Some Comments from a Numerical Analyst" (1970 Turing Lecture), Journal of the Association for Computing Machinery, Vol. 18, No. 2 (1971), pp. 137-147.

[Yp] T. Ypma, "Historical Development of the Newton–Raphson Method", SIAM Review, Vol. 37, No. 4 (1995), pp. 531–551.